Bộ môn An toàn Thông tin – Khoa Mạng máy tính và Truyền thông
Trường Đại học Công nghệ Thông tin – Đại học Quốc gia thành phố Hồ Chí Minh

1

Lab

PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

TỔNG QUAN KALI LINUX

Getting Comfortable with Kali Linux

Thực hành môn An toàn Mạng máy tính

Tháng 9/2022

Lưu hành nội bộ

<Nghiêm cấm đăng tải trên internet dưới mọi hình thức>

Mọi góp ý về tài liệu, vui lòng gửi về email inseclab@uit.edu.vn

BE SECURE
INSECLAB

A. TỔNG QUAN

1. Mục tiêu

- Giới thiệu hệ điều hành Kali Linux
- Quản lý các dịch vụ trên Kali Linux
- Các lệnh, công cụ thường sử dụng trên Kali Linux

2. Thời gian thực hành

- Thực hành tại lớp: **5** tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa **7** ngày.

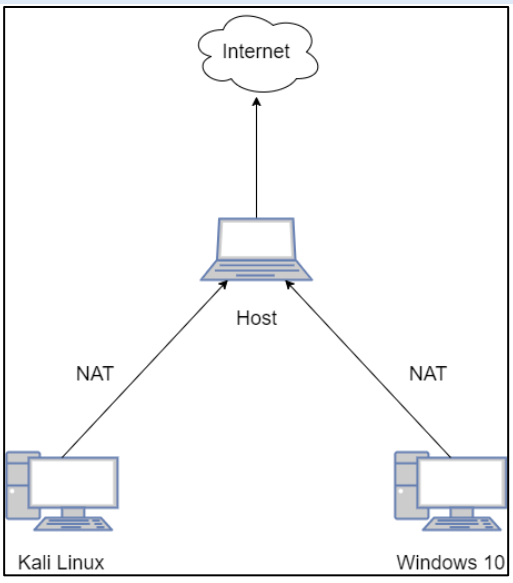
3. Kiến thức nền tảng

Kali Linux là một bản phân phối Linux (Linux distribution) dựa trên Debian được phát triển và duy trì bởi Offensive Security với mục đích tập hợp các công cụ bảo mật mã nguồn mở dành cho cho các nhà nghiên cứu bảo mật với các mục đích khác nhau như kiểm thử xâm nhập (penetration testing), pháp chứng (forensics), dịch ngược phần mềm (reverse engineering), ... Tất cả các công cụ đi kèm với hệ điều hành Kali Linux đã được đánh giá về tính phù hợp và tính hiệu quả của chúng bởi cộng đồng các nhà nghiên cứu bảo mật trên toàn thế giới.

4. Môi trường thực hành

Sinh viên cần chuẩn bị trước máy tính với môi trường thực hành như sau:

> Bài thực hành này sẽ sử dụng hai máy ảo Windows 10 và Kali Linux, chạy trong VMware Workstation (hoặc Virtualbox) theo mô hình mạng như hình bên dưới.



Hình 1. *Mô hình mạng bài thực hành*

B. CHUẨN BỊ MÔI TRƯỜNG

1. Cài đặt WMware Workstation

Hướng dẫn cài đặt, tạo máy và cấu hình máy ảo trên WMware Workstation trong tập tin PDF đính kèm: *HƯỚNG DẪN CÀI ĐẶT MÔI TRƯỜNG MÁY ẢO VỚI VMWARE.*

2. Các tập tin chuẩn bị

- Sinh viên có thể tải image Windows 10/11 tại https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/ và image Kali Linux tại https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/ .
- Sinh viên thiết lập cấu hình máy ảo như bảng bên dưới (hoặc có thể thay đổi tùy vào cấu hình của mỗi sinh viên. Tuy nhiên, card mạng vẫn phải giữ nguyên)

Máy ảo	Windows 10	Kali Linux
Dung lượng RAM	4GB	2 GB
Vi xử lý	1 Processor, 4 Cores	1 Processor, 4 Cor
Dung lượng ổ cứng	40 GB	80 GB
Card mạng	NAT (VMnet8)	NAT (VMnet8)

- Trên máy ảo Windows 10, tải các phần mềm sau: Nc.exe (https://eternallybored.org/misc/netcat/netcat-win32-1.12.zip)

C. THỰC HÀNH

1. Tổng quan Kali Linux

a) Hệ thống tập tin 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

b) Các lệnh Linux cơ bản

Bài thực hành 1: Liệt kê các tập tin

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.

```
root@kali:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
root@kali:~# ls /etc/apache2/sites-available/*.conf
/etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/default-ssl.conf
root@kali:~# ls -a1
.
..
.bash_history
.bashrc
.BurpSuite
.cache
.config
.dbus
```

Hình 2. *Liệt kê các tập tin/thư mục*

Bài thực hành 2: Di chuyển xung quanh

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.

```
root@kali:~# cd /usr/share/metasploit-framework/
root@kali:/usr/share/metasploit-framework# pwd
/usr/share/metasploit-framework
root@kali:/usr/share/metasploit-framework# cd ~
root@kali:~# pwd
/root
root@kali:~#
```

Hình 3. *Di chuyển xung quanh file system*

Bài thực hành 3: Tạo thư mục

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 **"_"**

```
root@kali:~# mkdir notes
root@kali:~# cd notes/
root@kali:~/notes# mkdir module one
root@kali:~/notes# ls
module  one
root@kali:~/notes# rm -rf module/ one/
root@kali:~/notes# mkdir "module one"
root@kali:~/notes# cd module\ one/
root@kali:~/notes/module one#
```

Hình 4. *Tạo thư mục trên Kali*

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.

```
root@kali:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
root@kali:~# mkdir -p test/{one,two,three}
root@kali:~# ls -1 test/
one
three
two
root@kali:~#
```

Hình 5. *Tạo cấu trúc thư mục*

c) Tìm kiếm tập tin trong Kali linux

Bài thực hành 4: which

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.

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

Hình 6. *Tìm kiếm tập tin sử dụng which*

Bài thực hành 5: locate

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**.

```
root@kali:~# updatedb
root@kali:~# locate sbd.exe
/usr/share/windows-resources/sbd/sbd.exe
root@kali:~#
```

Hình 7. *Tìm kiếm tập tin sử dụng lệnh locate*

Bài thực hành 6: find

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 8. *Tìm kiếm tập tin sử dụng lệnh find*

® Bài tập về nhà (yêu cầu làm)

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, KHÔNG thuộc sở hữu của user root và thực thi lệnh **ls -l** trên chúng. KHÔNG được sử dụng các lệnh pipeline/chaining*

2. 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.

Bài thực hành 6: Dịch vụ SSH

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 9. *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 10. *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 11. *Khởi động ssh cùng với hệ điều hành sử dụng lệnh systemctl*

Chúng ta có thể sử dụng lệnh **systemctl** để kích hoạt hoặc vô hiệu hóa hầu hết dịch vụ trên Kali Linux.

Bài thực hành 7: Dịch vụ HTTP

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 12. *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 13. *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 14. *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 15. *Liệt kê danh sách các dịch vụ có sẵn*

® **Bài tập về nhà (Cộng điểm)**

4. *Liệt kê các port đang được mở trên Kali Linux*
5. *Tại sao khi kiểm tra dịch vụ SSH có đang chạy hay không (Hình 10), kết quả hiển thị 2 dòng, trong khi dịch vụ HTTP (Hình 13), kết quả chỉ có 1 dòng.*
6. *Ngăn dịch vụ SSH chạy cùng với hệ thống lúc khởi động.*

3. Command line

d) Bash Environment

Bài thực hành 8: Biến môi trường

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 16. *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 17. *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 18. *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 19. *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.

```
root@kali:~# env
SHELL=/bin/bash
SESSION_MANAGER=local/kali:@/tmp/.ICE-unix/1300,unix/kali:/tmp/.ICE-unix/1300
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK_IM_MODULE=ibus
QT4_IM_MODULE=ibus
POWERSHELL_TELEMETRY_OPTOUT=1
SSH_AUTH_SOCK=/run/user/0/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=gnome
SSH_AGENT_PID=1217
GTK_MODULES=gail:atk-bridge
PWD=/root
LOGNAME=root
XDG_SESSION_DESKTOP=gnome
QT_QPA_PLATFORMTHEME=qt5ct
```

Hình 20. *Sử dụng lệnh env để hiển thị tất cả biến môi trường*

Bài thực hành 9: Bash history

Trong quá trình pentest, việc lưu lại lịch sử các lệnh đã được nhập rất quan trọng. Bash có lưu lại lịch sử của các lệnh đã được nhập, sử dụng lệnh **history.**

```
root@kali:~# history
    1  cat /etc/lsb-release
    2  clear
    3  history
root@kali:~#
```

Hình 21. *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)

```
root@kali:~# !1
cat /etc/lsb-release
DISTRIB_ID=Kali
DISTRIB_RELEASE=kali-rolling
DISTRIB_CODENAME=kali-rolling
DISTRIB_DESCRIPTION="Kali GNU/Linux Rolling"
root@kali:~#
```

Hình 22. *Sử dụng tiện ích history expansion*

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

```
root@kali:~# sudo systemctl restart apache2
root@kali:~# !!
sudo systemctl restart apache2
root@kali:~#
```

Hình 23. *Lập lại lệnh trước đó một cách dễ dàng*

® **Bài tập về nhà (Yêu cầu làm)**

7. *Lịch sử các lệnh thực ra được lưu trữ ở đâu? Liệt kê các ưu, nhược điểm khi thực hiện lưu trữ lại các lệnh đã nhập?*
8. *Có cách nào để ngăn chặn việc lưu trữ lịch sử lệnh hay không? Nếu có, hãy mô tả cách làm.*
9. *Ngoài cách sử dụng tiện ích history expansion, còn cách nào để thực hiện lại các lệnh đã nhập một cách nhanh chóng hay không? Nếu có, hãy mô tả cách làm.*

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.

Bài thực hành 10: 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 24. *Chuyển hướng kết quả vào tập tin*

Như trong Hình 25, 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.

Bài thực hành 11: 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 25. *Cung cấp tham số cho lệnh wc bằng toán tử <*

Bài thực hành 12: Chuyển hướng từ một tập tin

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 26. *Cung cấp tham số cho lệnh wc bằng toán tử <*

Bài thực hành 13: 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 27. *Chuyển hướng STDERR vào tập tin*

Theo Hình 28, 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*)

Bài thực hành 14: 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 28. *Piping kết quả của lệnh cat vào trong lệnh wc*

Trong Hình 29, 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

® Bài tập về nhà (Yêu cầu làm)

10. *Như đã biết, khi sử dụng toán tử ">" để xuất kết quả vô tập tin, nếu tập tin đã tồn tại, nội dung trong tập tin sẽ bị thay thế bằng nội dung mới. Vậy, có cách nào để hoàn tác lại quá trình này hay không? Nếu có, hãy mô tả cách làm.*
11. *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

Bài thực hành 15: grep

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 29. *Tìm kiếm bất kỳ tập tin nào trong /usr/bin có chứa chữ "zip"*

Bài thực hành 16: sed

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 30. *Thay thế từ trong output stream sử dụng lệnh sed*

Bài thực hành 17: cut

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 31. *Trích xuất các trường từ lệnh echo sử dụng lệnh cut*

Bài thực hành 18: awk

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 32. *Trích xuất các trường từ stream sử dụng lệnh awk*

® **Bài tập về nhà (Yêu cầu làm)**

12. *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.*

```
root@kali:~# YOUR COMMAND HERE
The user daemon directory is /usr/sbin
The user bin directory is /bin
The user sys directory is /dev
The user games directory is /usr/games
The user man directory is /var/cache/man
The user lp directory is /var/spool/lpd
The user mail directory is /var/mail
The user news directory is /var/spool/news
The user uucp directory is /var/spool/uucp
The user proxy directory is /bin
The user www-data directory is /var/www
The user backup directory is /var/backups
The user list directory is /var/list
The user irc directory is /var/run/ircd
The user gnats directory is /var/lib/gnats
The user nobody directory is /nonexistent
The user systemd-network directory is /run/systemd/netif
The user systemd-resolve directory is /run/systemd/resolve
The user _apt directory is /nonexistent
```

Hình 33. *Các thư mục home của user với shell là /usr/sbin/nologin*

13. *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.*

```
root@kali:~# YOUR COMMAND HERE
The IP Address [redacted] has hit [redacted]
The IP Address [redacted] has hit [redacted]
The IP Address [redacted] has hit [redacted]
The IP Address [redacted] has hit [redacted]
The IP Address [redacted] has hit [redacted]
The IP Address [redacted] has hit [redacted]
root@kali:~#
```

Hình 34. *Liệt kê danh sách địa chỉ IP cùng số lượng tương ứng*

g) Tải tập tin

Bài thực hành 19: wget

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

```
root@kali:~# wget https://github.com/blakduk/ahihi/raw/master/access_log.txt.gz -O access.txt.gz
--2020-08-16 13:20:36--  https://github.com/blakduk/ahihi/raw/master/access_log.txt.gz
Resolving github.com (github.com)... 13.250.177.223
Connecting to github.com (github.com)|13.250.177.223|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/blakduk/ahihi/master/access_log.txt.gz [following]
--2020-08-16 13:20:36--  https://raw.githubusercontent.com/blakduk/ahihi/master/access_log.txt.gz
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.8.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.8.133|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 3783 (3.7K) [application/octet-stream]
Saving to: 'access.txt.gz'

access.txt.gz          100%[=========================>]   3.69K  --.-KB/s    in 0.001s

2020-08-16 13:20:37 (5.97 MB/s) - 'access.txt.gz' saved [3783/3783]
```

Hình 35. *Tải xuống tập tin sử dụng lệnh wget*

Bài thực hành 20: curl

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.

```
root@kali:~# ls
Desktop    Downloads  Music       Pictures  redirection.txt  test
Documents  error.txt  output.txt  Public    Templates        Videos
root@kali:~# curl https://github.com/blakduk/ahihi/raw/master/access_log.txt.gz -o access.txt.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0   1289      0 --:--:-- --:--:-- --:--:--  1289
root@kali:~# ls
access.txt.gz  Documents  error.txt  output.txt  Public           Templates  Videos
Desktop        Downloads  Music      Pictures    redirection.txt  test
root@kali:~#
```

Hình 36. *Tải xuống tập tin sử dụng lệnh curl*

® Bài tập về nhà (Cộng điểm)

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

15. Theo bạn, trong 2 lệnh tải về wget và curl, lệnh nào ưu việt hơn? Giải thích?

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

4. Các công cụ cần thiết

h) Netcat

Netcat, được phát hành đầu tiên vào năm 1995 bởi *Hobbit*, là một trong những công cụ kiểm thử xâm nhập mạng (network penetration testing) "nguyên bản" và rất linh hoạt đến nỗi nó được tác giả ví như "con dao của quân đội Thụy Sĩ (Swiss army knife)" dành cho các hacker. Định nghĩa rõ ràng nhất về Netcat từ chính *Hobbit*: một tiện ích đơn giản giúp đọc và ghi dữ liệu qua các kết nối mạng, sử dụng giao thức TCP hoặc UDP.

Bài thực hành 21: Kết nối đến TCP/UDP port

Netcat có thể chạy dưới chế độ client hoặc server. Chúng ta sẽ bắt đầu với chế độ client trước tiên. Chúng ta có thể sử dụng chế độ client để kết nối tới bất kỳ cổng TCP/UDP nào, cho phép chúng ta:

- Kiểm tra xem port đang mở hay đóng.
- Kiểm tra banner từ dịch vụ đang lắng nghe trên port đó.
- Kết nối thủ công tới dịch vụ mạng.

Sử dụng lệnh **nc** để kiểm tra xem TCP port 110 (dịch vụ mail POP3) có đang mở hay không. Khai báo các tham số cho lệnh nc: **-v** để xem thêm chi tiết; địa chỉ IP (tên miền) muốn kết nối tới; và port muốn kết nối tới.

```
root@kali:~# nc -v mail.btopenworld.com 110
Warning: inverse host lookup failed for 213.120.69.88: Unknown host
mail.lb.btopenworld.com [213.120.69.88] 110 (pop3) open
+OK POP3 server ready.
```

Hình 37. *Sử dụng nc để kết nối tới một TCP port*

Hình 38 cho ta biết một số thông tin như sau: Đầu tiên, kết nối TCP thành công đến tên miền mail.btopenworld trên port 110, vì vậy Netcat thông báo port đích đã mở. Tiếp theo, máy chủ in ra thông điệp chào mừng, và yêu cầu đăng nhập, đây là hành vi bình thường của dịch vụ POP3.

Thử tương tác với máy chủ, thấy được chúng ta đã thành công trong việc nói chuyện với dịch vụ POP3 sử dụng Netcat (thậm chí đăng nhập thất bại).

```
root@kali:~# nc -v mail.btopenworld.com 110
Warning: inverse host lookup failed for 213.120.69.88: Unknown host
mail.lb.btopenworld.com [213.120.69.88] 110 (pop3) open
+OK POP3 server ready.
USER ahihi
+OK please send PASS command
PASS ahihi
-ERR Invalid user name or password
root@kali:~#
```

Hình 38. *Sử dụng nc kết nối tới dịch vụ POP3*

Bài thực hành 22: Lắng nghe trên TCP/UDP port

Thực hiện lắng nghe trên TCP/UDP port sử dụng Netcat rất có ích trong việc gỡ rối (debug) mạng của các ứng dụng client, hoặc để lắng nghe các kết nối mạng TCP/UDP. Thực hiện một chương trình chat đơn giản sử dụng Netcat như sau. Trên máy Kali, thực hiện mở 2 terminal có nhiệm vụ như sau:

- **Terminal 1:** Dùng để lắng nghe các kết nối TCP tới port 4444, sử dụng các tùy chọn **-n** để bỏ qua phân giải DNS, **-l** để sử dụng Netcat để lắng nghe kết nối, **-v** để hiển thị chi tiết và **-p** để chỉ định port lắng nghe

```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
```

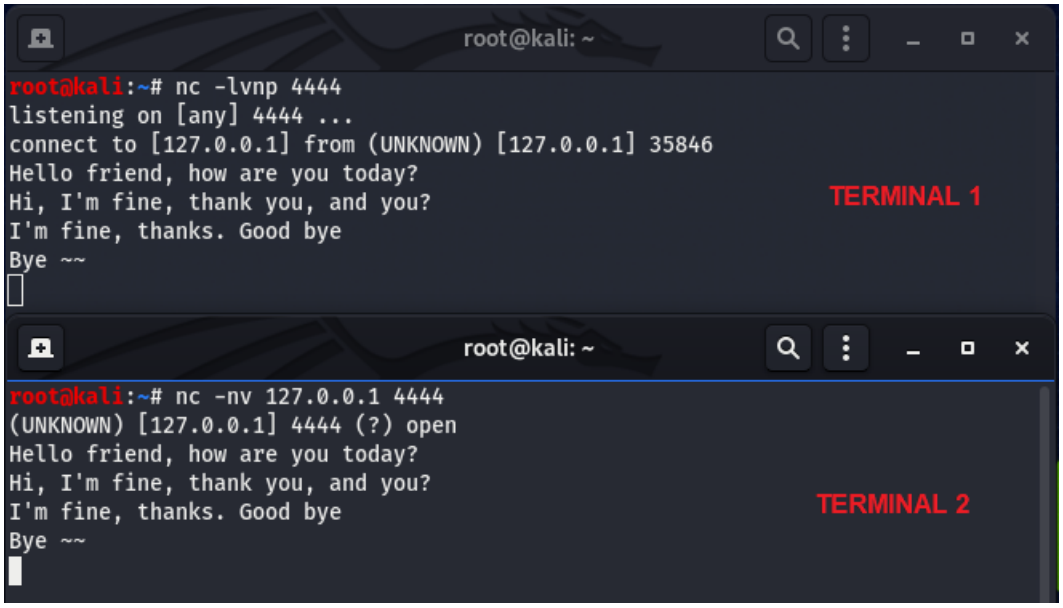
Hình 39. *Sử dụng nc để thiếp lập 1 nơi để lắng nghe kết nối*

- **Terminal 2:** Kết nối tới port 4444 trên chính máy của mình, sau đó thực hiện đánh nội dung bất kỳ.

```
root@kali: ~
root@kali:~# nc -nv 127.0.0.1 4444
(UNKNOWN) [127.0.0.1] 4444 (?) open
Hello friend, how are you today?
```

Hình 40. *Sử dụng nc để kết nối tới port 4444 trên chính máy kali*

Quan sát trên **Terminal 1** và thấy được nội dung vừa đánh trên **terminal 2**. Trên **Terminal 2**, thực hiện gõ lại và quan sát lại trên **Terminal 1**. Như vậy, một ứng dụng chat cơ bản sử dụng netcat đã được xây dựng thành công.

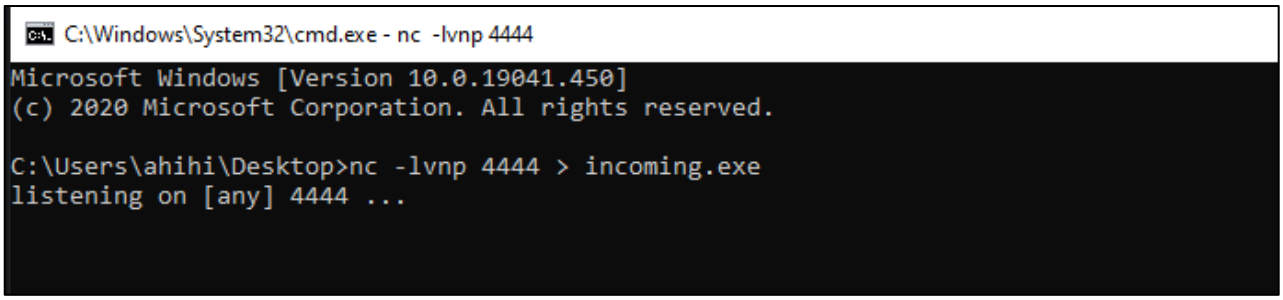


Hình 41. *Xây dựng một ứng dụng chat đang giản sử dụng netcat*

Bài thực hành 23: Trao đổi tập tin với Netcat

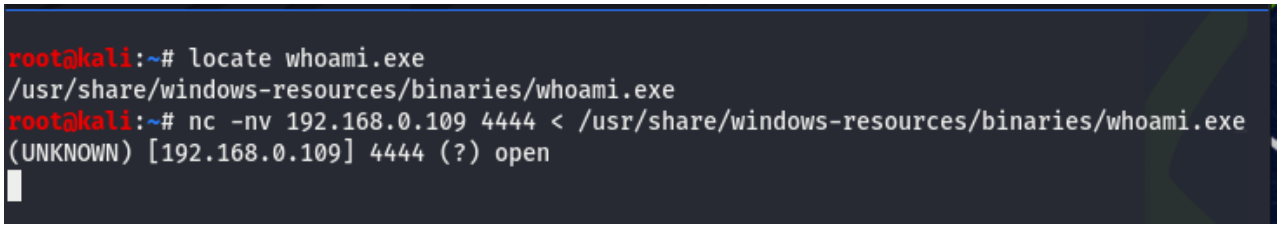
Netcat có thể được sử dụng để trao đổi tập tin, cả 2 định dạng là văn bản thô (text) và nhị phân (binary), từ máy này sang máy khác. Để gửi tập tin từ máy Kali sang máy Windows, chúng ta thực hiện triển khai giống với ứng dụng chat ở ví dụ trên, với một chút thay đổi.

- Trên máy Windows, chúng ta sẽ thiết lập lắng nghe trên port 4444 và chuyển tiếp kết quả vào tập tin có tên là **incoming.exe**



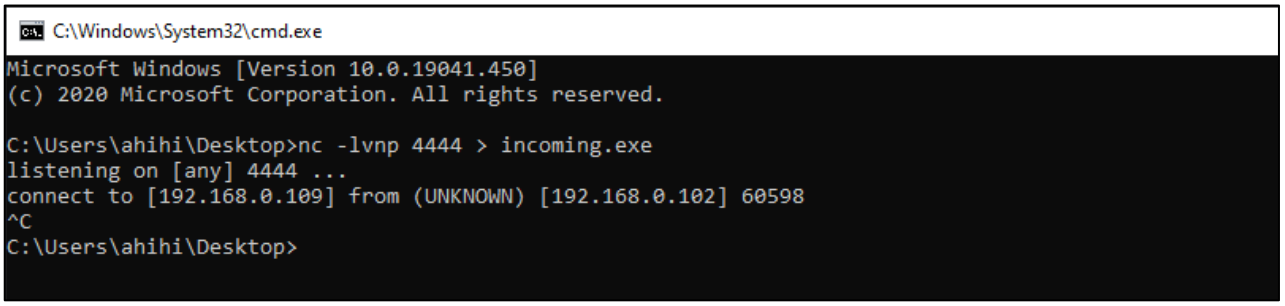
Hình 42. *Sử dụng nc để nhận tập tin*

- Trên máy Kali, chúng ta sẽ gửi tập tin **whoami.exe** lên máy Windows thông qua port 4444



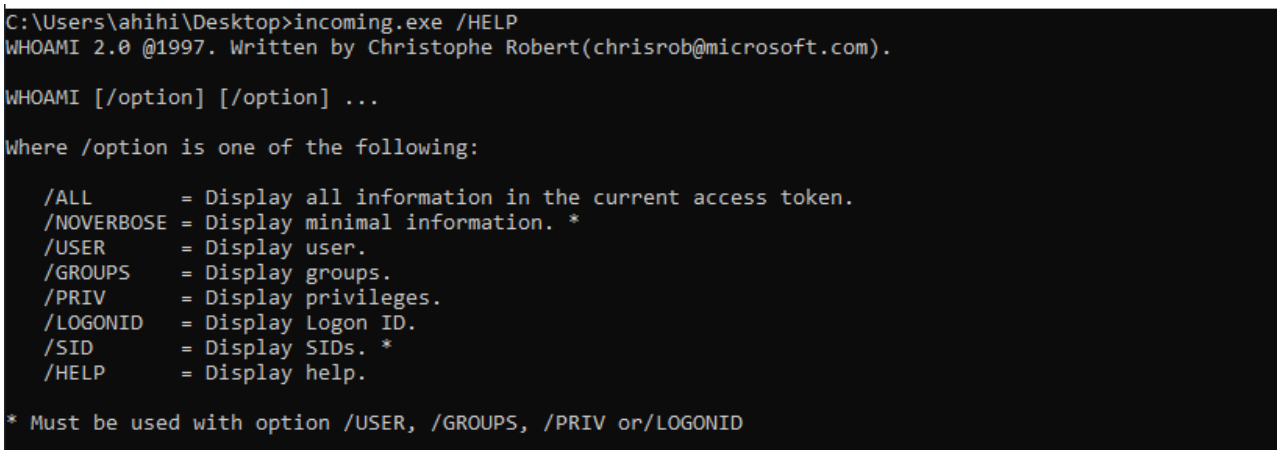
Hình 43. *Sử dụng nc để truyền file*

- Trên máy Windows, chungs ta nhận được kết nối từ máy Kali:



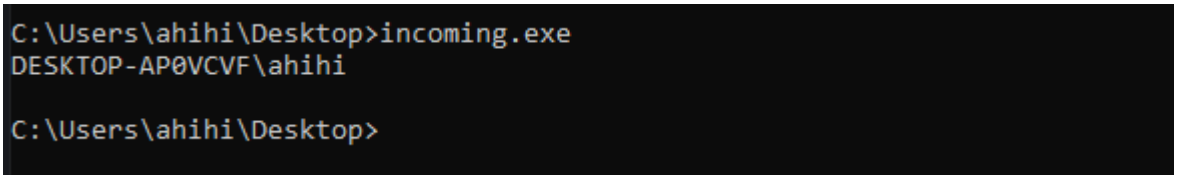
Hình 44. *Kết nối nhận được trên máy Windows*

Lưu ý rằng, chúng ta sẽ không nhận được bất kỳ thông báo nào về quá trình trao đổi tập tin. Trong trường hợp trên, vì tập tin truyền từ máy Kali sang máy Windows có kích thước nhỏ, không tốn nhiều thời gian. Để kiểm tra, chạy tập tin **incoming.exe** trên máy Windows.



Hình 45. *Thực thi tập tin được gửi thông qua nc.*

Kết quả sau khi thực thi tập tin này, cho chúng ta biết được tên user (lệnh tương tự như **whoami** trên Windows)



Hình 46. *Kết quả trả về sau khi thực thi tập tin incoming.exe*

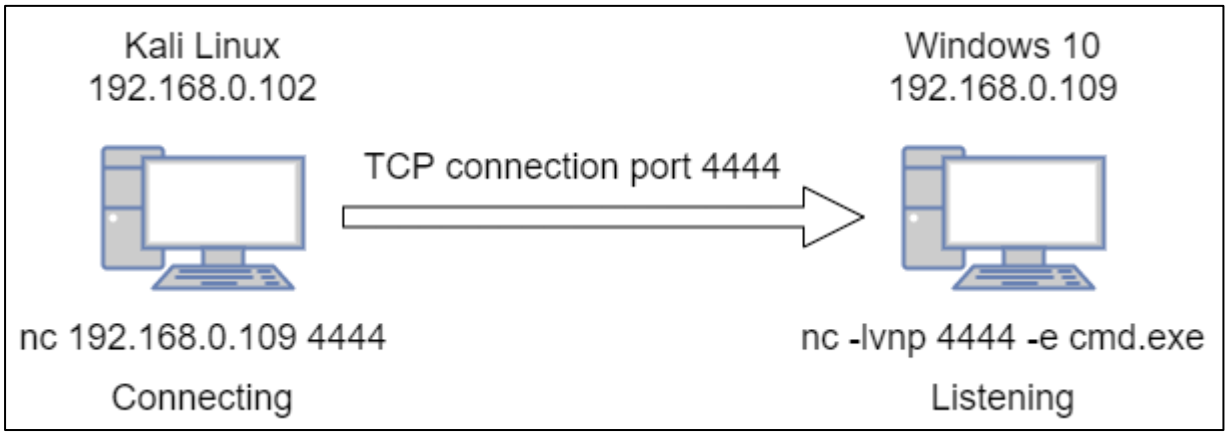
Bài thực hành 24: Quản trị từ xa với Netcat

Một trong những tính năng hữu ích của Netcat là khả năng chuyển hướng lệnh. Bằng cách sử dụng thuộc tính **-e**, netcat sẽ thực thi một chương trình sau khi thực hiện hoặc nhận được kết nối thành công. Dưới góc độ bảo mật, tính năng này mở ra nhiều khả năng thú vị, và do đó sẽ không có sẵn trong hầu hết các hệ thống Linux/BSD hiện đại. Tuy nhiên, do Kali Linux là một bản phân phối dành cho các pentester, vì vậy phiên bản Netcat trên Kali vẫn hỗ trợ thuộc tính -e.

Khi được kích hoạt, thuộc tính này có thể chuyển hướng input, output và các thông điệp lỗi trong quá trình thực thi vào TCP/UDP port thay vì xuất lên màn hình mặc định.

Ví dụ, khi thực thi **cmd.exe**, bằng cách chuyển hướng các stdin, stdout và stderr vào mạng. Do đó, mọi kết nối tới cổng mà đang tiến hành chạy **cmd.exe**, chúng ta đã có thể thực thi các lệnh của máy từ xa.

Bind Shell sử dụng Netcat

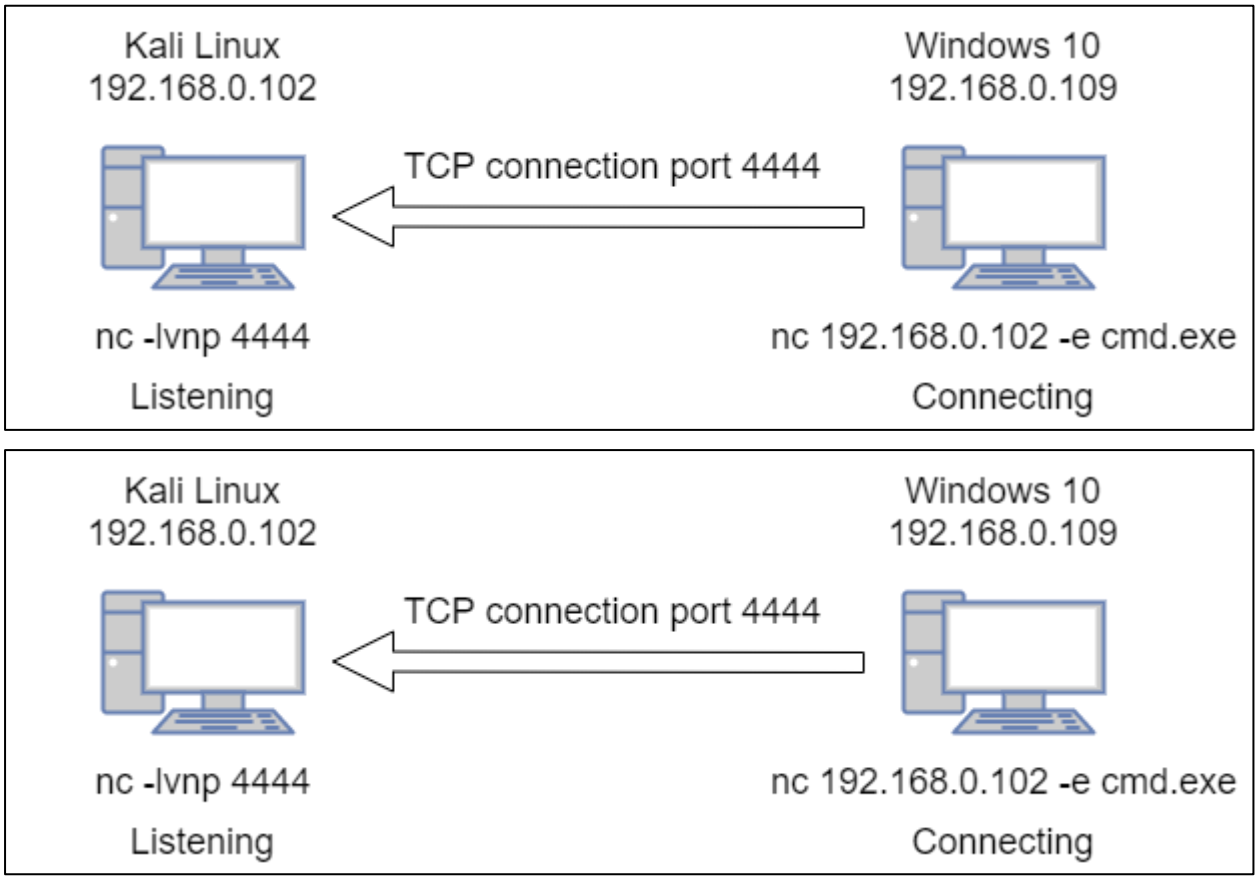


Hình 47. *TCP Bind Shell sử dụng netcat*

Giả sử máy Windows 10 là máy nạn nhân, máy Kali Linux sẽ là máy của kẻ tấn công. Bind shell là một session shell mà trong đó, máy nạn nhân sẽ thực hiện lắng nghe trên port (do kẻ tấn công tạo ra), và chờ kẻ tấn công kết nối vào port đó để thực hiện điều khiển máy từ xa.

Ở ví dụ trên Hình 48, bằng cách sử dụng thuộc tính **-e** của Netcat, máy nạn nhân sẽ thực hiện chạy chương trình **cmd.exe** trên **port 4444**. Khi kẻ tấn công kết nối vào máy nạn nhân trên port 4444, kẻ tấn công có thể thực thi được các lệnh command prompt trên máy nạn nhân.

Reverse Shell sử dụng Netcat



Hình 48. *TCP Reverse Shell sử dụng netcat*

Ngược lại với Bind Shell, bây giờ, kẻ tấn công sẽ thực hiện lắng nghe trên một port bất kỳ. Máy nạn nhân sẽ kết nối vào port đó và cung cấp cho kẻ tấn công shell của mình để kẻ tấn công có thể điều khiển máy nạn nhân từ xa.

Ở ví dụ trên Hình 49, kẻ tấn công sẽ thực hiện lắng nghe trên **port 4444**. Máy nạn nhân sẽ thực hiện kết nối vào port 4444 trên máy kẻ tấn công và cung cấp chương trình **cmd.exe** (của máy nạn nhân) để kẻ tấn công có thể điều khiển từ xa.

® Bài tập về nhà (Yêu cầu làm)

Triển khai ứng dụng chat đơn giản trên 2 máy Kali và Windows 10. Và trả lời các câu hỏi sau:

17. *Máy chủ nào sẽ đóng vai trò là server?*
18. *Máy chủ nào sẽ đóng vai trò là client?*
19. *Nếu khai báo lệnh "nc -lvnp 4444" thì thật chất, port 4444 được mở ở máy nào?*
20. *Thực hiện chuyển tập tin wget.exe trên máy Kali sang máy Windows 10.*
21. *Thực hiện lại chi tiết kịch bản Reverse Shell và Bind Shell sử dụng netcat.*
22. *So sánh ưu và nhược điểm khi sử dụng Reverse Shell và Bind Shell? Khi nào nên sử dụng Bind Shell? Khi nào nên sử dụng Reverse Shell?*

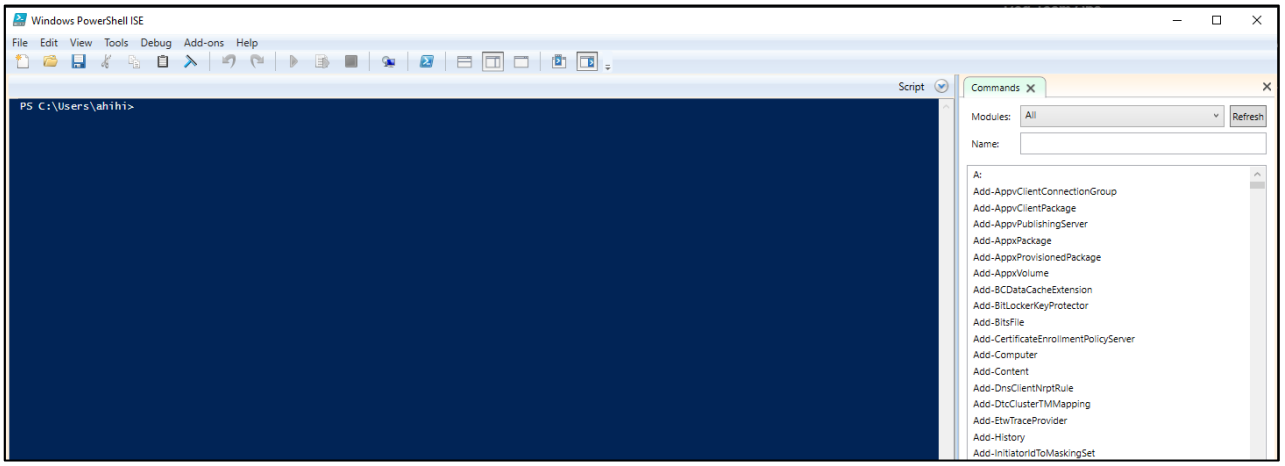
i) PowerShell

Windows PowerShell được thiết kế đặc biệt cho quản trị viên hệ thống và người dùng tự động hóa nhanh chòng việc quản trị nhiều hệ điều hành (Linux, MacOS, Unix và Windows) và các tiến trình liên quan đến các ứng dụng chạy trên chúng.

Tóm lại, PowerShell là một công cụ có thể phục vụ cho việc kiểm thử xâm nhập và có thể được cài đặt trên (hoặc đã được cài đặt mặc định) các phiên bản Windows khác nhau. Nó được cài đặt mặc định trên các nền tảng Windows hiện đại bắt đầu từ Windows Server 2008 R2 và Windows 7.

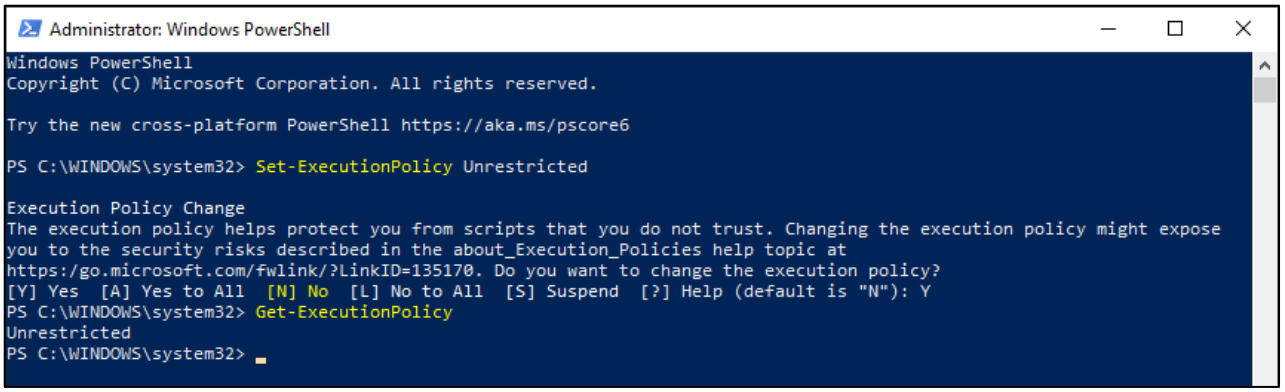
- Windows PowerShell 5.0 chạy trên các phiên bản Windows sau:
- Windows Server 2016/Windows 10 (được cài đặt mặc định)
- Windows Server 2012 R2/Windows Server 2012/Windows Server 2008 R2 SP1/Windows 8/Windows 7 SP1 (cần cài đặt Windows Management Framework 5.0 để có thể chạy được)
- Windows PowerShell 4.0 chạy trên các phiên bản Windows sau:
- Windows 8.1/Windows Server 2012 R2 (được cài đặt mặc định)
- Windows 7 SP1/Windows Server 2008 R2 SP1 (cần cài đặt Windows Management Framework 4.0 để có thể chạy được)
- Windows PowerShell 3.0 chạy trên các phiên bản Windows sau:
- Windows 8/Windows Server 2012 (được cài đặt mặc định)
- Windows 7 SP1/Windows Server 2008 R2 SP1 (cần cài đặt Windows Management Framework 3.0 để có thể chạy được)

PowerShell chứa một môi trường phát triển tích hợp (Integrated Development Environment – IDE) được tích hợp sẵn, được gọi là Windows PowerShell ISE (Integrated Scripting Environment). ISE là một host application dành cho PowerShell cho phép chạy các lệnh, viết, kiểm tra, và debug các tập lệnh trong một giao diện đồ họa người dùng (graphical user interface – GUI).



Hình 49. *PowerShell ISE*

PowerShell duy trì một chính sách thực thi nhằm xác định loại script PowerShell nào được chạy trên hệ thống. Chính sách mặc định là "Restricted", có nghĩa là hệ thống sẽ không tải các tập tin cấu hình PowerShell cũng như chạy các script PowerShell. Trong bài lab này, chúng ta sẽ thiết lập chính sách thành "Unrestricted" bằng cách chạy PowerShell dưới quyền Administrator (Run As Administrator), sau đó nhập lệnh **Set-ExecutionPolicy Unrestricted**



Hình 50. *Thiết lập chính sách thực thi trên PowerShell*

PowerShell có nhiều công dụng cho phép chúng ta thực hiện nhiều tác vụ mà không cần phải cài đặt thêm công cụ trên máy của đối tượng. Cũng giống như Netcat, powershell cũng cung cấp các tính năng phục vụ cho kiểm thử xâm nhập như:

- Trao đổi tập tin
- Bind Shell
- Reverse Shell

® Bài tập về nhà (Cộng điểm)

23. Thực hiện trao đổi tập tin, bind shell và reverse shell sử dụng PowerShell

24. Ngoài netcat và powershell, còn cách nào có thể tạo ra được reverse shell và bind shell không? Cho một ví dụ.

D. YÊU CẦU & ĐÁNH GIÁ

1. Yêu cầu

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.. Đăng ký nhóm cố định từ buổi 1.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng **1 trong 2 hình thức**:

j) Báo cáo chi tiết:

Báo cáo cụ thể quá trình thực hành (có ảnh minh họa các bước) và giải thích các vấn đề kèm theo. Trình bày trong file PDF theo mẫu có sẵn tại website môn học.

~~**k) Video trình bày chi tiết:**~~

~~Quay lại quá trình thực hiện Lab của sinh viên kèm thuyết minh trực tiếp mô tả và giải thích quá trình thực hành. Upload lên **Youtube** và chèn link vào đầu báo cáo theo mẫu. **Lưu ý:** *Không chia sẻ ở chế độ Public trên Youtube.*~~

Đặt tên file báo cáo theo định dạng như mẫu:

[Mã lớp]-LabX_MSSV1-Tên SV1_MSSV2 –Tên SV2

Ví dụ*: [NT101.I11.1]-Lab1_14520000-Viet_14520999-Nam.*

- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.

2. Đánh giá:

- Sinh viên hiểu và tự thực hiện được bài thực hành, đóng góp tích cực tại lớp.
- Báo cáo trình bày chi tiết, giải thích các bước thực hiện và chứng minh được do nhóm sinh viên thực hiện.
- Hoàn tất nội dung cơ bản và có thực hiện nội dung *mở rộng – cộng điểm* (với lớp **ANTN**).

Kết quả thực hành cũng được đánh giá bằng kiểm tra kết quả trực tiếp tại lớp vào cuối buổi thực hành hoặc vào buổi thực hành thứ 2.

Lưu ý: Bài sao chép, nộp trễ, "*gánh team*", ... sẽ được xử lý tùy mức độ.

HẾT

Chúc các bạn hoàn thành tốt!