Lecture 2: Getting comfortable with Linux

BKACAD's Security Training

Table of Content

Booting up Kali Linux

Install

- Vmware Workstation
- Kali Linux 64-bit (amd64)
  - https://images.offensive-security.com/virtual-images/kali-linux-2020.2-vmware-amd64.7z

Booting up Kali Linux

Login

- Username: kali
- Password: kali

Booting up Kali Linux

Running as root permission

Many commands will require elevated privileges to run, fortunately, the sudo command can overcome this problem.

Booting up Kali Linux

Kali Menu

https://tools.kali.org/

Booting up Kali Linux

Exercises

1. Boot your Kali operating system and change the kali user password to something secure
2. Take some time to familiarize yourself with the Kali Linux menu
3. Using the Kali Tools site, find your favorite tool and review its documentation. If you don’t have a favorite tool, pick any tool
4. Try to use “Snapshot” in VMware

Linux Filesystem

Kali Linux adheres to the filesystem hierarchy standard (FHS) which provides a familiar and universal layout for all Linux users. The directories you will find most useful are:

- /bin - basic programs (ls, cd, cat, etc.)
- /sbin - system programs (fdisk, mkfs, sysctl, etc)
- /etc - configuration files
- /tmp - temporary files (typically deleted on boot)
- /usr/bin - applications (apt, ncat, nmap, etc.)
- /usr/share - application support and data files

Basic Linux Commands

Man Page - man

```
kali@kali:~$ man ls
```

Basic Linux Commands

Listing Files - ls

```
kali@kali:~$ ls

kali@kali:~$ ls /etc/apache2/sitesavailable/*.conf

kali@kali:~$ ls -a1
```

Basic Linux Commands

Creating Directories - mkdir

```
kali@kali:~$ mkdir notes

kali@kali:~$ cd notes/

kali@kali:~/notes$ mkdir module one

kali@kali:~/notes$ ls

module one

kali@kali:~/notes$ rm -rf module/ one/

kali@kali:~/notes$ mkdir "module one"

kali@kali:~/notes$ cd module\ one/

kali@kali:~/notes/module one$
```

Basic Linux Commands

Finding in Linux - which

The which command searches through the directories that are defined in the $PATH environment variable for a given file name.

```
kali@kali:~$ which python

/usr/bin/python
```

Basic Linux Commands

Finding in Linux - locate

In order to provide a much shorter search time, locate searches a built-in database named locate.db rather than the entire hard disk itself.

```
kali@kali:~$ sudo updatedb
```

```
kali@kali:~$ locate shell.exe
```

Basic Linux Commands

Finding in Linux - find

The find command is the most complex and flexible search tool

```
kali@kali:~$ find starting/path options expression
action
```

```
kali@kali:~$ find /home -name *.jpg
```

Basic Linux Commands

Exercises

1. Use man to look at the man page for one of your preferred commands.
2. Use which to locate the pwd command on your Kali virtual machine.
3. Use locate to locate wce32.exe on your Kali virtual machine.
4. Use find to identify any backup file (.bak) at ~ and execute ls -l on them.

   Chaining/piping commands is NOT allowed!

Basic Linux Commands

Managing Kali Linux Services

HTTP & SSH

```
kali@kali:~$ sudo service apache2|ssh start

kali@kali:~$ sudo systemctl enable apache2|ssh
```

Basic Linux Commands

Exercises

1. Practice starting and stopping various Kali services.
2. Enable the HTTP service to start on system boot.

Basic Linux Commands

Searching, Installing, and Removing Tools

```
kali@kali:~$ apt update

kali@kali:~$ apt upgrade

kali@kali:~$ apt-cache search x

kali@kali:~$ apt-install x

kali@kali:~$ apt remove --purge x

kali@kali:~$ dpkg -i x.deb
```

Basic Linux Commands

Exercises

1. Search for a tool not currently installed in Kali
2. Install the tool
3. Remove the tool
4. Revert Kali virtual machine to previously taken snapshot (optional)

Command Line Fun

compgen

The commands that user can run are either files, built-ins or key-words

```
kali@kali:~$ compgen -k

kali@kali:~$ compgen -b

kali@kali:~$ compgen -c
```

Command Line Fun

Tab Completion

The Bash shell auto-complete function allows us to complete filenames and directory paths with the tab. This feature accelerates shell usage so much that it is sorely missed in other shells.

```
kali@kali:~$ ls D[tab]
```

Command Line Fun

Bash History Tricks

Bash maintains a history of commands that have been entered, which can be displayed with the history command.

```
kali@kali:~$ history
```

Command Line Fun

Bash History Tricks

One of the simplest ways to explore the Bash history is right from the command line prompt. We can browse through the history with some useful keyboard shortcuts with the two most common being:

- scroll backwards in history
- scroll forwards in history

Command Line Fun

Downloading Files

```
kali@kali:~$ wget -O report_wget.pdf https://www.offensive-security.com/reports/penetration-testing-sample-report-2013.pdf
kali@kali:~$ curl -o report.pdf https://www.offensive-security.com/reports/penetration-testing-sample-report-2013.pdf
```

Command Line Fun

Exercises

1. Download the PoC code for an exploit from https://www.exploit-db.com using curl, wget, saving each download with a different name.

Command Line Fun

Editing Files from the Command Line

```
kali@kali:~$ touch filename.txt

kali@kali:~$ nano filename.txt

kali@kali:~$ vi filename.txt

kali@kali:~$ gedit filename.txt
```

Command Line Fun

Piping and Redirection

Every program run from the command line has three data streams connected to it that serve as communication channels with the external environment

- `stdin`
- `stdout`
- `stderr`

Command Line Fun

Piping and Redirection

Piping (using the | operator) and redirection (using the > and < operators) connects these streams between programs and files to accommodate a near infinite number of possible use cases

Command Line Fun

Piping and Redirection

```
kali@kali:~$ handon < data.in > results.out 2> err.mes

kali@kali:~$ handon < data.in > results.out 2>&1 # err.mes combine with normal output

kali@kali:~$ handon < data.in &> results.out # shorter than light

kali@kali:~$ handon < data.in > /dev/null # Discard all outputs

kali@kali:~$ handon < data.in | tee results.out # Display output and save it to
results.out

kali@kali:~$ handon < data.in | tee -a results.out # Display output and append it to
results.out

kali@kali:~$ handon < data.in >> results.out # Append stdout to file

kali@kali:~$ handon < data.in &>> results.outs # Append both stdout and stderr to file
```

Command Line Fun

Running Commands in the Background

```
kali@kali:~$ ping 127.0.0.1 > ping.log &

kali@kali:~$ ping 127.0.0.1 &> ping.log &

kali@kali:~$ jobs # List any tasks currently running in the background

kali@kali:~$ fg <job_number> # The corresponding job number to bring task back to foreground

kali@kali:~$ # If task in fg, press Ctrl+Z to suppend the process and then

kali@kali:~$ bg # to continue the process in the background
```

Command Line Fun

Exercises

1. Write a command that executes ifconfig and redirects standard output to a file named ipaddress.txt.
2. Write a command that executes ifconfig and redirects standard output and appends it to a file named ipaddress.txt
3. Write a command that performs a directory listing (ls) on the root file directory and pipes the output into the more command.
4. Write a command that executes mytask.sh and sends it to the background.
5. Given the following job list, write the command that brings the Amazon ping task to the foreground

```
[1]   Running          ping www.google.com > /dev/null &
[2]-  Running          ping www.amazon.com > /dev/null &
[3]+  Running          ping www.oreilly.com > /dev/null &
```

Command Line Fun

Process Control

```
kali@kali:~$ ps -ef

kali@kali:~$ kill pid
```

Command Line Fun

File monitoring - tail

The most common use of tail is to monitor log file entries as they are being written. The -f option (follow) is very useful as it continuously updates the output as the target file grows

```
kali@kali:~$ tail -f /var/log/apache2/access.log
```

Command Line Fun

grep

Search the content of the files for a given pattern and prints any line where pattern is matched. Provide it with a pattern and one or more file names.

```
kali@kali:~$ ls -la /usr/bin | grep zip
```

Command Line Fun

sed

sed performs text editing on a stream of text, either a set of specific files or standard output.

```
kali@kali:~$ sed 's/regexp/replacement/flags'
kali@kali:~$ echo "I need to try hard" | sed 's/hard/harder/'
```

Command Line Fun

cut

It is used to extract a section of text from a line and output it to the standard output. Some of the most commonly-used switches include -f for the field number we are cutting and -d for the field delimiter.

```
kali@kali:~$ cut -d ":" -f 1 /etc/passwd
```

Command Line Fun

awk

Same as cut, a commonly used switch with awk is -F, which is the field separator, and the print command, which outputs the result text.

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

Command Line Fun

Exercises

1. Start the Firefox browser on your Kali system. Use ps and grep to identify Firefox's PID.
2. Terminate Firefox from the command line using its PID.
3. Start your apache2 web service and access it locally while monitoring its access.log file in real-time.

Command Line Fun

Exercises

1. Using /etc/passwd, extract the user and home directory fields for all users on your Kali machine for which the shell is set to /bin/false. Make sure you use a Bash one-liner to print the output to the screen.

Output:

```
The user mysql home directory is /nonexistent
The user Debian-snmp home directory is /var/lib/snmp
The user speech-dispatcher home directory is /var/run/speech-dispatcher
The user Debian-gdm home directory is /var/lib/gdm3
```

Command Line Fun

Exercises

1. Using curl to download Apache log file from http://www.offensive-security.com/pwk-files/access_log.txt.gz
2. Use Bash commands to inspect the file and discover various pieces of information, such as who the attackers were and what exactly happened on the server. (use cat, cut, sort, uniq and grep commands)

Command Line Fun

Exercises

1. Using wget to download DVWA's source code from https://github.com/ethicalhack3r/DVWA/archive/master.zip
2. With only grep commands, find any PHP files that contain system( ) function in DVWA's source code. (using regular expression with –e option)