LINUX(kali,ubuntu,etc)

IGNORE ALL CAPITALIZATION OF SYNTAX

Binaries This term refers to files that can be executed, similar to executables in Windows. /usr/bin or /usr/sbin

Case Sensitivity

Directory

Home /home

Kali Kali Linux is a distribution of Linux specifically designed for penetration testing.

Root

Scrip

CLI – command line interface is where type and see output of our command

Shell – software environment for running commands in linux. Most used shell is bash (Bourne again shell)

Terminal is a command line interface(software the shell program runs inside of).

THE LINUX FILESYSTEM

/ ---- IS THE ROOT. The following are subdirectories in the root.

/root – home directory of root user

/etc - linux configuration file

/home – home directory for user

/mnt - Where other filesystems are attached or mounted to the filesystem

/media Where CDs and USB devices are usually attached or mounted to the filesystem

/bin Where application *binaries* (the equivalent of executables in Microsoft Windows) reside

/lib Where you'll find *libraries* (shared programs that are similar to Windows DLLs)

/dev /proc /sys – Kernel and system information

BASIC COMMANDS IN LINUX

GENERAL COMMAND SYNTAX

|FOLLOWS BELOW|

COMMAND OPTIONS ARGUEMENT

Clear - clear screen

Df- disk utilization file system

Apropos – used. For search for command you don't know e.g apropos list, spropos write, apropos "search for files"

Pwd – tell you where you at

Whoami- see which user you're logged in as

cd – change directory

cd .. (used to move up one level in directory**)**

- You would use .. to move up one level.
- You would use .. .. to move up two levels.
- You would use .. .. .. to move up three levels, and so on.

Cd / - move to root level (/)

Ls – list content of a directory

Ls – l (list files,owner,permission,and size)

Ls -la (list files that are hidden)

--help (used to get help) e.g. python –help

-h (used to get help) e.g. python -h

-? (used to get help)

Man – used to describe manual page of a command e.g man python3.

locate – used to find stuff e.g locate python3

whereis – used to locate the binary file e.g. whereis python3

which- returns the location of the binaries in the PATH variable in Linux. E.g. which python3

find - The find command is the most powerful and flexible of the searching utilities.

Syntax is (find directory options expression)

Find . – name "poe*" – finds all name that starts with poe in current directory

Find ~/documents -name “*d*” – find from document directory all that starts with d or before d

Find / -type f – name apache2. (First, I state the directory in which to start the search, in this case /υ. Then I specify which type of file to search for, in this case f for an ordi- nary fileϖ. Last, I give the name of the file I’m searching for, in this case apache2.**)**

find /etc -type f --name apache2.* (using a * wild card to find every apache2.anything)

WILDCARD - ?,*,[] ,

?at – look for any single word + at e.g fat,cat,bat,hat

Rm poems?.txt – only removes poems with any single number after the poems

[c,b] – look for words start with c and b followed by at eg cat, bat

*** - list unlimited words that ends with at e.g what, mnat,hjkjdsat ,etc. e.g**

Mv *.txt newpath - move all txt files to newpath

Mv oldpath/* . -moves all file in oldpath to current path

Grep - you can use the grep command as a filter to search for keywords.

Eg grep “the” poems.txt – it highlights all the “the” in peom

Grep -n “the” poems.txt – hightlight the “the” in peom and also print in lines numbers\

Grep -in “the” poems.txt – hightlight all the “the” “The” regardless of case sensitive in poem

Grep -vi “the” poems.txt – omit all “the” or “The” in peoms txt

Grep -E “[hijk]” poems.txt – regular expresions for ccurrences of h,I,j,k

Grep -E “\w{6,}” poems.txt – prints all character of words 6 or more

Ps – used to display processes running on the machine

Aux – ps followed by aux to display process information

Piping – we use the command **|** (allow us to take output of one command and send it as an input to another command)

e.g ps aux | grep apache2

cat – used to create smaller file and it can also be used to display a file , the cat command followed by the filename eg **cat** > **hackingskills** (to create short file, then type short words, use ctrl D or ctrl C to exit) **cat hackingskills** (displays the wors in the file).

Touch – used to create a file or touch existing file, however if no file exist it create a new one eg touch file

Mkdir – creates new directory eg mkdir dir

Can also create multiple directory eg mkdir life/case life/great

Create in parent directory eg mkdir -p life/case/life

Cp – to copy file eg. Cp oldfile newfile

Rmdir – to remove empty directory eg rmdir dir2

Rm -r - remove nonempty directory eg rm -r direc2

Rm – to remove file eg rm file

Using wild card ?

Rm poems?.txt – only removes poems with any single number after the poems

Mv – to move file or also rename eg mv fromfile tofile

To rename e.g mv oldname newname

To move to current directory using the (.) e.g mv oldfile .

Using wildcard

Mv *.txt newpath - move all txt files to newpath

Mv oldpath/* .   -moves all file in oldpath to current path

CHAPTER 2

Head

Head snort.conf – view first line of the document

Head -n5 poems.txt – read the first 5 lines of the text

Head -20 snort.conf – go to first 20 lines of the text

Tail

Tail -n3 poems.txt

Tail snort.conf – view last line of the document

Tail. -20 snort.conf – view last 20 lines of the text

Nl snort.conf – nl is used to display lines number in the text #####

Cat

Cat snort.conf | grep output – shows all the lines that has output in snort text

Cat poems.txt | cat – n | tail -n5 – output the pems into number lines and print out five lines of the tail

Cat poems.txt | tail -n5 | cat -n – output peoms into five lines at the tail and then print out the lines

awk – used to extract specific test from a file according to a rule

eg awk ‘{print $2}’ l.txt – prints the second colomn

awk ‘{print $2 “\t” $1}’ l.txt – prints both second and first colomn

awk ‘{print $2 “\t” $1}’ l.txt | sort -n – prints both second and first colomn sorted

Sed - lets you search for occurrences of a word or a text pat- tern and then perform some action on it by replacing

e.g sed s/mysql/MySQL/g /snort.conf > snort2conf. (you want sed to replace every occurrence of *mysql* with *MySQL* (remember, Linux is case sensitive) and then save the new file to *snort2.conf*.)

s and g means all

s and 2 means replace second occurrence e.g sed s/mysql/MySQL/2 snort.conf >snort2.conf

g only means the first occurrence e.g sed s/mysql/MySQL/ snort.conf >snort2.conf(repleces only first occurence)

more - displays a page of a file at a time and lets you page down through it using the enter key

less - less command is very similar to more, but with additional functionality —hence, the common Linux aficionado quip, “Less is more.” You can search for stuff when you press / and type n to move to next file

eg less poems.txt

sort – sort the files eg sort tx.txt

sort k2 -n tx.txt – sort based on second colomun starting from file line

sort -u tx.txt – sort for unique files, removing duplicate

rev – prints text in reverse sequence

tac – concatenates or displays file in reverse

tr- translates or modifies individual characters according to parameters

nano /etc/hostname – to change your hostname

sudo useradd -m username – to add new user for your linux

sudo passwd username – to add the password

```
sudo usermod -a -G sudo username – to add username to sudo grouo

sudo chsh -s /bin/bash username – change the bash shell
```

file – tells us type of file eg file document

stat – tells us details of a file

wc -l – tells word count lines

ABSOLUTE PATH ,RELATIVE PATH, TILDE EXPANSION

ABSOLUTE PATH – begins from the root of the file system eg. /home/scott/Documents

RELATIVE PATH - begins from the current working directory eg Documents

The ".." refers to the parent directory of current working directory eg

working directory -> /home/scoot
relative path -> ..
result-> /home
TILDE EXPANSION(~) – refers to the current user's home directory e.g ~/document -> /home/scoot/document

NAVIGATING A FILE SYSTEM

To move file without space

Cd exercise – move to exercise folder.

To move file with space, you use back slash after each word or put words in quotes.

Cd exercise\ files\

Cd “exercise files”

To see what is in a directory recursively.

Ls -R /departments

To switch back to previous directory I was working on.

Cd –

To switch back to home directory

Cd

SUPERUSER PRIVILEGES

Sudo - to use the root user privileges eg sudo ls /root

Sudo -k – to give up the privileges

Sudo -s – to change from normal user to root user

PERMISSION

File Permission – R-read W-write X-execute

Rwxrwxrwx file1

User – the first three rwx

Group – the next three rwx

Others – the last three rwx

Changing file permission

Chmod – change permission mode string

Chown and chgrp – change the files owners group

Methods to represent permission

-octal (755, 644 and 777)

User - Read(4) write(2) execute(1) = total 7

Group- Read(4) write(–) execute(1) = total 5

Others – read(4) write(-) execute(-) = total 4

Rwxr-xr—

-symbolic (a=r, g+w and o-x)

(+) = add permission, (-) = remove permission, (=) = resets permission

User(u) – read(+) write(+) execute(+) = u+rwx

Group(g) – read(=) write() execute() = g=r

Others(o)= read(-) write() execute() = o-rwx

All(a) = read(=) write(=) execute(=) = a=rwx

Modify file permission

To remove user permission from execute

Chmod 644 test.sh

Chmod -x test.sh

To remove user permission from read

Chmod u-r test.sh

Chmod 244 test.sh

To change ownership

Sudo chown root test.sh – changed the file to root user and cant edit except with root

Sudo chown toby test.sh - changed the file to toby user and cant edit except with root

LINKS

Links are files that rference other files. Use to avoid having duplicate files

Types

Hard Link – points to specific data (by inode) on the disk

Ln poems.txt words.txt

Soft Link or Symbolic link(symlink) – points to another file

    ln -s poems.txt writing.txt

vim – text editor eg. Vim ti.txt

nano – text editor eg nano ti.txt

TAR ARCHIVE

Tar -cvf myfiles.tar exercise\ files/ - c creates, v verbose(list files), f output to a files , this is create uncompressed tar files

Tar -czf myfiles.tar.tgz exercise\ files/ - creates a tar files

Tar -caf myfiles.tar.gz exercise\ files/ - creates compression of the tar file

Tar -xf myfiles.tar.bz2 – this extract the tar file.

Tar -xvf logtar.tar.gz – extract this file

Tar -xf myfiles.tar.gz -C unpack2 – to extract files into directory unpack2

ZIP AND UNZIP

Zip -r .zip exercise\ files/ - create a zip files

Unzip exfiles.zip – unzip your zip files

Unzip exfiles.zip -d unpack4 – this unzip into a directory unpack4

REDIRECTION

Stdin – 0 standard input

Stdout – 1 standard output

    Ls 1> files.txt –output content in file

    Ls > files.txt

Stderr – 2 standard error

Ls notreal 2> files.txt –output error

>> - used to append eg echo "hello" >> files.txt

PATH

PATH – location which shell search for executables programs e.g echo $PATH

Editing the $PATH variable – edit the shell profile(~/.bash_profile)

INFORMATION OF YOUR SYSTEM

Ls -l /etc/*release – shows all in release

Cat /etc/*release – output our version

Uname -a – shows name of system and our kernel

Free -h – shows memory of machine

Cat /proc/cpuinfo – shows your cpu info

Df -h – shows the disk space on system

Sudo du -hd1 / - shows all the space used on my whole system, du- disk usage, h -size, d-details, 1 -1 level deep from root

Sudo lshw | less – shows what hardware or devices attached, less let its hows lesser
Ip a – networking information

INSTALLING AND UPDATING SOFTWARE

Debian(ubuntu,mint,etc) – use apt
Red Hat and CentOS – use yum or DNF
Fedora – use DNF
SUSE – use YaST
Arch – uses pacman

SEARCH
On ubuntu, use – apt search tree (shows all software with tree)
To install – sudo apt install tree
To update and upgrade all my software packages.

```
Sudo apt update
Sudo apt upgrade
```

ADVANCED LINUX

KERNEL

To find which kernel you on. – cd /boot -changed to where kernel is

- Uname – r – shows you the type of linuz kernel

Command for hardwares –

- lshw and lspsci, lsusb and lsbk, lscpu and lsdev

Hardware control – hdparm
Output and input – inb and oub
Configure - setpci

System calls

System calls are implemented by the kernel and meant to be called from user space
Include/uapi/asm-generic/uinstd.h
Standard Library uses architecture-dependent means to invoke system call mechanism

Printk() is the kernel function for code to print messages.
Dmesg – shows RAM buffer message from kernel

- Dmesg
- Dmesg | wc -l (list total lines of dmesg)
- Dmesg | grep command (tells you how kernel boots)
- Journalctl -t kernel (list stuff on rams etc)
- Journalctl -t kernel | grep command (tells you how many times system booted)
- Journalctl -t kernel -f (shows current activity)

Surveying the linux kernel

Proc and sysfs filesystems are virtual filesystems
Proc filesystems is mounted on /proc at boot
Proc is for process info
Sysfs filesystem is mounted on /sys at boot.
Sysfs is for kernel object info
Device files – character and block drivers use device files.