

# Basic Linux Commands and Capabilities

## 1 Overview

The objective of this lab is to introduce the students to the basic Linux commands that will be used in FIT3031 unit. Tasks in this lab can be tried on any Unix/Linux OS. But we encourage you to do all tasks in any Ubuntu docker container in SecureCorp GNS3 project.

## 2 Lab Tasks

Once you start all nodes in the project open the terminal of any node (Internal-Attacker, Internal-Client etc.) and run the below commands. All nodes in SecureCorp project runs a minimized version of Ubuntu. Below commands will help to install few packages required for this lab.

```
unminimize
apt install man-db
apt install python3
```

### 2.1 General Unix commands

Familiarize yourself with the usage of the following commands and complete the task with the guidance of your tutor.

<code>man &lt;cmd&gt;</code>	an interface to the on-line reference manuals
<code>whatis &lt;cmd&gt;</code>	display manual page descriptions
<code>&lt;cmd&gt; --help</code>	print a help message and exit.
<code>history</code>	prints the last 50 commands
<code>grep</code>	print lines matching a pattern
<code>cat</code>	concatenate files and print on the standard output
<code>&lt;cmd&gt; &gt; file</code>	send the output of the command to that file

Table 1: General commands

1. Which text-based command provides information on the use of other Linux commands and utilities? Identify at least two.
2. View the man page of the command `grep`. Export the output to a file and see the content of the exported file.

```
man grep > how_to_use_grep
cat how_to_use_grep
```

3. Identify the methods of running a command that you ran a short while ago without re-typing it. Use it to re-run the cat command from previous question.

```
history
!<line_number_of_the_command_from_history_list>
```

4. Which key-stroke invokes auto-completion of commands and filenames?

### 2.2 Basic networking commands

Use man pages to understand the usage of the below commands and complete the task with the guidance of your tutor.

<code>ifconfig</code>	configure a network interface
<code>ping</code>	send ICMP ECHO_REQUEST to network hosts
<code>netstat</code>	print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships
<code>hostname</code>	show or set the system's host name
<code>route</code>	show / manipulate the IP routing table

Table 2: Networking commands

1. Check the IP address of the node. You can use one of the following commands.

```
ip address
ifconfig
```

2. Check the Hostname of the node.

```
hostname
```

3. Check the local routes in the node and the current network connections in the node.

```
netstat
route
```

4. Check the connectivity with the internet by pinging the Google DNS server.

```
ping 8.8.8.8
```

## 2.3 File and Directory manipulation

### 2.3.1 Root Directory and its Sub-directories.

While we are poking around the Linux file system, take a look at the files in the root directory. You should see directories with names like `/bin`, `/home`, `/lib`, and `/usr`.

Every file and directory in the file system has a path. Unix paths are delimited by forward slashes (`/`).  
e.g. `/home/username/fit3031/`

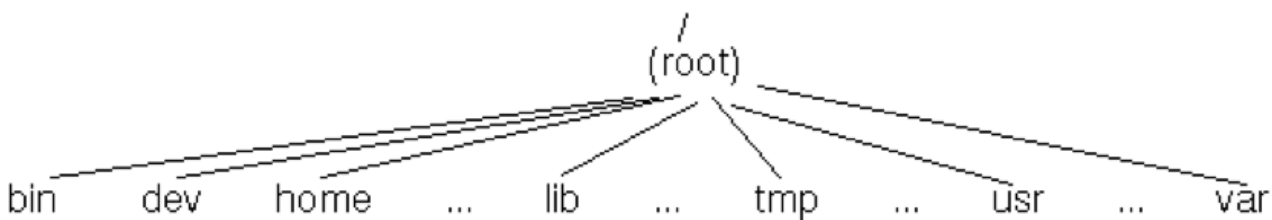


Figure 1: Unix file system hierarchy.

### Special directories.

- Root Directory ( `/` ): The Top-Most directory.
- Bin Directory: Executable programs that comprise the GNU/Linux utilities.
- Home Directory ( `~` ): Current User's directory.
- Current Directory ( `.` ): The directory you're in.
- Parent Directory ( `..` ): The directory above.
- Absolute paths: start with ( `/` ). i.e the root directory.
- Relative paths: start from your current directory.

<code>cat &lt;file name&gt;</code>	concatenate files and print on the standard output
<code>mkdir</code>	make directories
<code>rmdir</code>	remove empty directories
<code>rm</code>	remove files or directories
<code>cp</code>	copy files and directories
<code>mv</code>	move or rename files and directories
<code>find</code>	search for files in a directory hierarchy
<code>locate</code>	find files by name
<code>pwd</code>	print location of current/working directory
<code>ls</code>	list directory contents
<code>cd</code>	change working directory
<code>chmod</code>	change file mode bits
<code>tar</code>	The GNU version of the tar archiving utility

Table 3: File and directory manipulation commands

### 2.3.2 File and Directory manipulation

- Find your current working directory.  
`pwd`
- Change your working directory to /home  
`cd home`
- Create a new directory with the name 'scripts'  
`mkdir scripts`
- Create a new file in the scripts directory with the name rsa.py and copy the content from the textbook-RSA.py file in Moodle. Save the file.  
`nano scripts/rsa.py`  
Press CTRL+X to save and exit
- Rename the rsa.py file as textbook-rsa.py  
`mv scripts/rsa.py scripts/textbook-rsa.py`
- List the permissions of the file. Give everyone read and executable permissions to the file. Only the owner (root) should have write permissions  
`ls -l scripts`  
`chmod 755 scripts`
- Run the textbook-rsa.py file using the relative path as well as the absolute path of the file.  
`python3 scripts/textbook-rsa.py`  
`python3 /home/scripts/textbook-rsa.py`
- Find all 'print' statements in the textbook-rsa.py file without opening the file.  
`grep 'print' scripts/textbook-rsa.py`
- Copy the textbook-rsa.py file to /home  
`cp scripts/textbook-rsa.py ./textbook-rsa.py`
- Delete the /home/scripts/textbook-rsa.py file.  
`rm scripts/textbook-rsa.py`
- Remove the scripts directory.  
`rmdir scripts`

## 2.4 Linux Shell

You can write programs to do all the things you want in the Unix shell. A Shell script is a bunch of commands saved in one executable file with extension `.sh`.

1. What is your default Linux shell?

```
echo $SHELL
```

2. What is the PATH variable, what does it do, and how do you inspect its value?
3. Create a simple shell script `script.sh` for listing all files and sub directories in the `/etc` directory.

```
nano script.sh  
ls -l /etc  
Press CTRL+X to save and exit
```

4. Make your script executable and run the script.

```
chmod +x script.sh
```

5. Why it is necessary to use the prefix `./` explicitly to run an executable file?