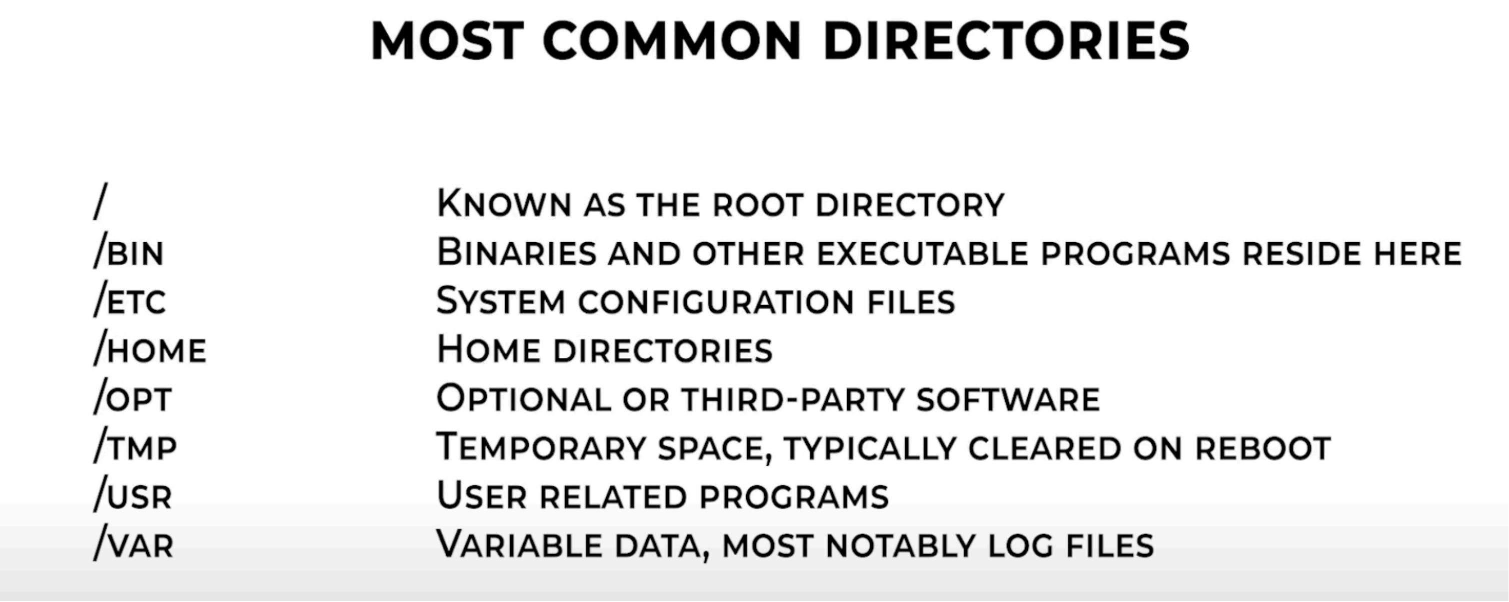
About Kali Linux

Kali Linux *(formerly known as BackTrack Linux)* is an open-source, Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. It does this by providing common tools, configurations, and automations which allows the user to focus on the task that needs to be completed, not the surrounding activity.

Kali Linux contains industry specific modifications as well as several hundred tools targeted towards various Information Security tasks, such as Penetration Testing, Security Research, Computer Forensics, Reverse Engineering, Vulnerability Management and Red Team Testing.



MOST COMMON DIRECTORIES

/	Known as the root directory
/bin	Binaries and other executable programs reside here
/etc	System configuration files
/home	Home directories
/opt	Optional or third-party software
/tmp	Temporary space, typically cleared on reboot
/usr	User related programs
/var	Variable data, most notably log files

⚠️ Everyday commands list:

1. hostname - to find out the hostname
   -i to get the ip address of the host
   -f to get long hostname

2. ip addr - to get the ip address
   -4 followed by addr
   -6 followed by addr
   To get ipv4 or ipv6 ip address

3. ip route
   To know who your default router is or if I've overridden any routes.

4. whois google.com – Getting Information on Domain Name
   whois 8.8.8.8 – Getting Information about IP Address.
   whois command is a utility for retrieving information about a domain or an IP address.
   # alternate command - dig, nslookup

5. route

   Show or manipulate the IP routing table

6. iwconfig

   If you have an external USB, you can use the iwconfig command to gather crucial information for wireless hacking such as the adapter's IP address, its MAC address, what mode it's in, and more.

7. ifconfig eth0 192.168.181.115

   To change your IP address, enter ifconfig followed by the interface you want to reassign and the new IP address you want assigned to that interface.
   For example, in a denial-of-service
   (DoS) attack, you can spoof your IP so that that the attack appears to come from another source, thus helping you evade IP capture during forensic analysis. This is a relatively simple task in Linux, and it's done with the ifconfig command.

   Changing Your Network Mask and Broadcast Address

   ifconfig eth0 192.168.181.115 netmask 255.255.0.0 broadcast 192.168.1.255

8. netstat - to get network activity

   -a to display all sockets
   -l to display listening server sockets
   -u to display UDP ports
   ss command is modern replacement of netstat and faster

9. last

   The last command shows the last logins on the system. This can sometimes help to detect any unusual login activity.

10. ping - to test connectivity and system availability. **Packet InterNet Groper**

    Ping works using the ICMP protocol
    Explanation: Sends ICMP echo requests to a specified IP address to check network connectivity and measure round-trip time.
    Example: Running ping 8.8.8.8 would send ICMP echo requests to the IP address "8.8.8.8" (Google's DNS server) and display the round-trip time and packet loss statistics.
    ping -c 5 www.google.com - Control number of packets sent
    ping -i 3 google.com - Changing the time interval of next ping. Here 3 sec
    ping -c 5 -q google.com - to get only a summary. Here summary of 5 packets

ping -w 5 google.com - to set timeout of pinging. Here it will stop on 5th

11. `nmcli` - nmcli is a command-line tool for controlling NetworkManager

```
nmcli connection show
nmcli -p  device show
```

-p, –pretty: This option prints the output in an organized format which is convenient and easily readable to humans.

To find out IP's under the network

```
#!/bin/bash

if [ "$1" == "" ]
then
echo "You forgot your IP address"
echo "Syntax: ./ipsweep.sh 192.168.4"

else
for ip in `seq 1 254`; do
ping -c 1 $1.$ip | grep "64 bytes" | cut -d " " -f 4 | tr -d ":" &
done
fi
```

12. `whereis python`

If you're looking for a binary file, you can use the whereis command to locate it.

13. `ifconfig eth0 down`

```
ifconfig eth0 hw ether 00:11:22:33:44:55
ifconfig eth0 up
```

Spoofing Your MAC Address by changing it

To spoof your MAC address, simply use the ifconfig command's down option to take down the interface (eth0 in this case). Then enter the ifconfig command followed by the interface name (hw for hardware, ether for Ethernet) and the new spoofed MAC address. Finally, bring the interface back up with the up option for the change to take place.

14. `dirb http://10.10.123.87`

Dirb is a command line tool you can use to fuzz websites or web apps. Dirb finds files and directories on your target site that are not directly linked from a publicly accessible page on the site or from the Internet.

15. `netdiscover -r 192.168.71.142/24`

    The netdiscover command is a tool that gathers information about a network. It can be used to find potential IP addresses on a network, and to gather information about connected clients and routers.

    -r scan a given range instead of auto scan

    The command above will scan all the available IPs

16. `curl  https://example-files.online-convert.com/document/txt/example.txt`

    This will display the content of the url data

    `curl -O  https://example-files.online-convert.com/document/txt/example.txt`

    This will download the file from the URL

    -O means output

17. `script logfile.log`

    It will capture and grow as we use terminal for perform activities.

18.
    ```
    sudo getent shadow
    sudo getent shadow username
    sudo getent shadow username username
    sudo getent passwd
    sudo getent group
    sudo getent services
    sudo getent hosts
    sudo getent networks
    ```

    getent is a Unix command that helps a user get entries in a number of important text files called databases. This includes the passwd and group databases which store user information – hence getent is a common way to look up user details on Unix.

Managing Users and Groups

1. `cat /etc/passwd`

   This command will show all the existing users
2. `cat /etc/passwd | wc -l`

   This will display the total number of users

3. su username
   To switch user, if using root put sudo at the beginning
4. sudo sysadminctl -addUser khan21 -password khan interactive
   To create a user in Mac terminal
5. sudo adduser username
   To give privilege to the new user
6. sudo useradd username
   Will create user only, without password and additional info
7. sudo usermod -g marketing khan
   Changing the group for khan to marketing
8. sudo usermod -l mmhk khan
   This will now change the login name of the user khan to mmhk
9. userdel
   To delete user from the system
10. sudo groupadd sales
    Here I am creating a group called sales, an id will be automatically generated for it
11. cat /etc/group
    Will display all the existing group in the system.
12. sudo groupdel sales
    It will delete the group sales
13. sudo groupmod sales -n marketing
    Here I am renaming the group sales into marketing, The group id remain the same
14. sudo gpasswd -a sales khan
    Here I am adding user khan to the group sales using -a
    Now if you check cat /etc/group then you will see the user khan added next to group name
15. sudo gpasswd -d sales khan
    If you want to take user khan out of the sales group using -d
16.
17. sudo chown -R khan Documents/
    This will chenge only the owner to khan over everything the Documents dir has
18. sudo chown -R khan:sales Documents/
    This will not only change the owner to khan also the group to sales as well
19.

⚠️ find command

```
find /home/Movies
```

Will fild all the files and folder under /home/Movies

```
find . -maxdepth 1
```

It will only search current directory and only level 1 folder

```
find /home -type f -name sales.txt
```

Find files based on filename

-type f means file

-name means name of the file

```
find /root -type f -size 11c -name khan
```

Find files based on size

-size to mention size(c for bytes, k for kilobytes, m for megabytes, g for   gigabytes

Here it will find any file name khan and size 11 bytes

```
find / -type f -size +100k
```

Will look for any fild bigger than 100 kilobytes

If you want less than use -100k

You can use -size multiple times if you want search file bigger than N number and less than N number

```
find /home -type d -name pictures
```

Find Directory based on directory name

```
find /etc/server -type f -user john
```

Find files based on username

```
find / -type f -newermt ‘6/30/2020 0:00:00'
```

/ means root directory and it will search every files and folder

Find files modified after a specific date

(all dates/times after 6/30/2020 0:00:00 will be considered a condition to look for)

```
sudo find ~/Desktop -type f -size +100k -size -5M -exec cp {} ~/Desktop/copy_here \;
```

This command will search any file under desktop between 100k to 5m will copy them to ~/Desktop/copy_here folder

-exec flat will replace {} with the paths and cp command will copy them to the destination

To end this command we need \;

```
sudo find ~/Desktop -type f -size +100k -size -5M -ok cp {} ~/Desktop/copy_here \;
```

This is same as before except it will ask you Y/N that you want to copy or not

```
find / -type f \( -name 8V2L -o -name bny0 -o -name c4ZX -o -name D8B3 -o -name FHl1 -o -name oiMO -o -name PFbD -o -name rmfX -o -name SRSq -o -name uqyw -o -name v2Vb -o -name X1Uy \) -exec ls {} -ilrt
```

```
\; 2>>/dev/null
```

Here the command looking for 12 different files and listing them with details information

Systemctl command:

The systemctl command in Linux is a utility that manages and interacts with the systemd system and service manager. The systemd is a system initialization system and a service manager that has become the default on many Linux distributions.

- sudo systemctl start/stop/status <service_name>
  To start, stop, or check status of a service

- sudo systemctl restart <service_name>
  To restart the service

- sudo systemctl enable <service_name>
  # Enabling a service causes the system to start the service upon reboot or whenever a computer starts up.
  # The enable subcommand does not start the particular service immediately.
  # You need admin privilege.

- sudo systemctl enable --now sshd
  To enable and start at the same time

- sudo systemctl is-enabled sshd
  To check if a service is enabled

- sudo systemctl disable <service_name>
  To disable a service
  One can manually start the service
- sudo systemctl mask <service_name>
  To prevent the service to start

- sudo systemctl unmask <service_name>
  To unmask a service to be able to start

- systemctl list-sockets
  systemctl --show-types list-sockets --all

- systemctl | grep -i Running | more
  To filter out what services are running. more command will control the display

⚠️ **Command help:**

1. man ls - will display the manual of any command
2. ls --help - will display all the details of ls command
3. whatis ls - will display short description of ls command

⚠️

Create file:

1. touch filename.txt
2. cat > filename.txt
3. echo “Hello this is a file” > filename.txt
4. printf “Hello this is a file” > filename.txt
5. nano filename.txt
6. vi filename.txt
7. vim filename.txt
8. gedit

MISC commands

- ps - report a snapshot of the current processes

  ```
  ps -U root -u root u
  ```

  To see every process running as root (real & effective ID) in user format

  ```
  ps aux
  ps aux | grep apache2
  ```

  The ps aux Linux command is a commonly used command in Linux for obtaining information about running processes.

  ```
  pstree
  ```

  Display a tree of processes
  pstree can have PID number followed by pstree command

- Verify the current bash

  ```
  ps $$
  ```

  To switch bash

  ```
  exec bash/zsh
  ```

- To go root user

  ```
  > sudo su
  > exit  - to exit root user
  ```

- The passwd command is used to change the password for the current user.

  ```
  > passwd
  ```

- Change ssh keys

  ```
  cd /etc/ssh
  ls
  sudo mkdir old-keys
  sudo mv ssh_host_* old-keys/
  sudo dpkg-reconfigure openssh-server
  ```

  # This is so you can make sure that you have secret keys that are not being used by other users that have downloaded the same VM.

  Setup ssh

  ```
  sudo apt-get install openssh-server
  ```

  # to install openssh

  ```
  sudo service ssh start/stop  - to start/stop the ssh service
  sudo service ssh status  - to check service status
  ```

  ```
  ssh -V
  ```

  Locally check version

  ```
  ssh -V root@10.10.246.160
  ```

Get to know the verion of ssh of remote server

```
ssh username@192.168.1.141
```

To connect machine using ssh encrypted protocol

{Problem:

```
Unable to negotiate with 192.168.1.142 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss
```

The error message "no matching host key type found" indicates that the client and server cannot agree on the host key algorithm type. This can happen when using ssh from OpenSSH >= 8.8 and images that use an older ssh server version.

Solution:

```
ssh -oHostKeyAlgorithms=+ssh-dss username@192.168.1.141
```

}

If you want to crackRSA private key's passphrase

Step 1:

```
python ssh2john.py id_rsa > id_rsa.hash
```

ssh2john.py is part of John the Reaper tool
id_rsa was the actual RSA file
We are writing it to id_rsa.hash

Step 2:

```
john id_rsa.hash -wordlist=rockyou.txt
```

Using john tool we are decoding the id_rsa.hash file
-wordlist option requires a wordlist, here we use rockyou.txt

```
grep aws data.txt
```

This will find the word aws in data.txt file

```
grep -n aws data.txt
```

This will find the word aws along with the line number in data.txt file

```
grep -c aws data.txt
```

This will find total number of occurrences of the word aws

```
ls /home/khan | grep user
```

The pipe (|) sign indicate multiple command

The command will perform the first command and then forward the output to the next command grep user. Now grep will find all the files and directories inside /home/khan that starts with the word user.

Regular Expression to Match IP Addresses

```
grep -E -o "([0-9]{1,3}[\.]){3}[0-9]{1,3}" file.txt
```

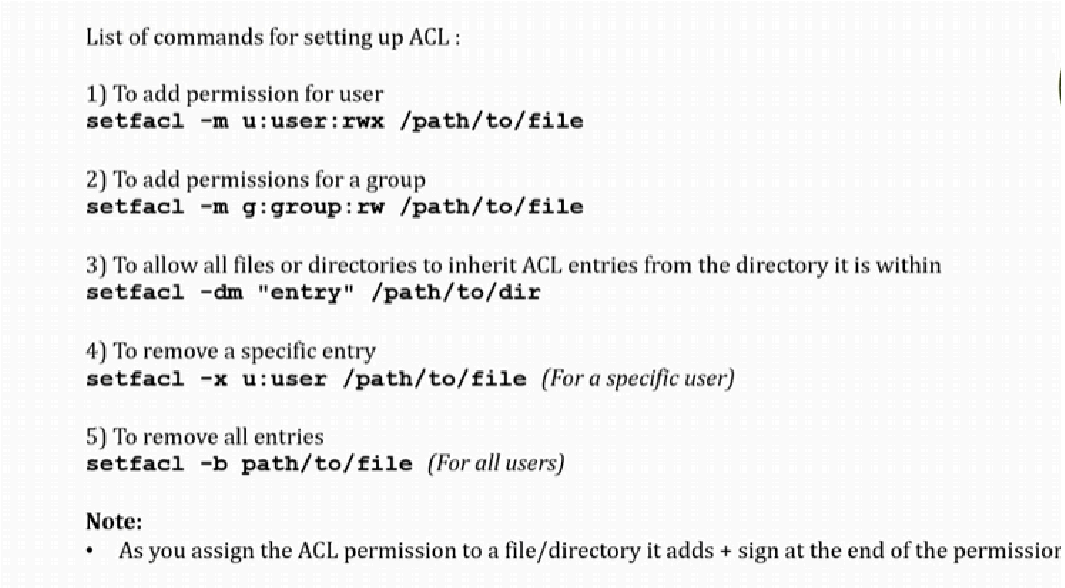
Match only Valid IPv4 Addresses

```
grep -E -o
"(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[
0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" file.txt
```

-E, –extended-regexp
-o, –only-matching

Access Control List(ACL)

An access control list (ACL) is made up of rules that either allow access to a computer environment or deny it. In a way, an access control list is like a guest list at an exclusive club. Only those on the list are allowed in the doors.



List of commands for setting up ACL :

1) To add permission for user
```
setfacl -m u:user:rwx /path/to/file
```

2) To add permissions for a group
```
setfacl -m g:group:rw /path/to/file
```

3) To allow all files or directories to inherit ACL entries from the directory it is within
```
setfacl -dm "entry" /path/to/dir
```

4) To remove a specific entry
```
setfacl -x u:user /path/to/file
```
(For a specific user)

5) To remove all entries
```
setfacl -b path/to/file
```
(For all users)

Note:

- As you assign the ACL permission to a file/directory it adds + sign at the end of the permissior

```
touch test.txt
ls -l test.txt
 -rw-r--r-- 1 kali kali 396 Feb 25 16:46 test.txt
getfacl test.txt
# file: test.txt
# owner: kali
# group: kali
user::rw-
group::r--
Other::r–

setfacl -m u:sales:rw /tmp/test.txt
```

Using setfacl you can apply permissions
-m means modify
u:sales:rw, u means user: followed by username:followed by read and write permission
If you want apply the changes to the group just put g in place of u

A cron expression is a simple tool used to automate the scheduling of tasks such as database updates, batch processing, or regular system maintenance. Data engineers, system admins, IT professionals, and other software engineers commonly use cron expressions to streamline repetitive tasks.

```
crontab -e
```

Will let you choose what editor you want to use and select them by their number. It will become the default for the next time unless you change using export editor=nano; crontab -e or by editing nano .selected_editor under home directory

Six section of CRON job
m h dom mon dow   command

# m - minute, h - hour, dom - day of month, mon - month, dow - day of week, command is the actual script to you write to execute

```
crontab -l
```

List the crontab entries

```
crontab -r remove
```

Remove the crontab

```
crond
```

Crontab deamon/service that manages scheduling

```
* * * * * echo "Hello cron" >> ~/Desktop/hello.txt
```

Here:
* for minute means run every minute(if specific type: 10 or 5 or 25 etc)
* for hour means run every hour(if specific tyle: 12 or 15 etc as it follows 24 hours system)
* for day of month(dom) every day(if specific type: 1 or 6 or 10 etc)
* for month means every month(if specific type: JAN, FEB, JUN etc)
* for day of week means every week day(if specific type: SUN, MON, THU etc)
And followed by the command:
echo "Hello cron" >> ~/Desktop/hello.txt

Note: you can add as many cron job in a single crontab

```
15  * * * * echo "Hello cron" >> ~/Desktop/hello.txt
```

Here this job will run every 15 minutes past the hour like(10:15, 12:15, 2:15 etc)

```
15 11 * * * echo "Hello cron" >> ~/Desktop/hello.txt
```

Here this job will run everyday at 11:15 am

```
15 11 10  * * echo "Hello cron" >> ~/Desktop/hello.txt
```

Here this cron job will run only 10th day each month at 11:15 am

```
15 11 10  JUN * echo "Hello cron" >> ~/Desktop/hello.txt
```

Here this cron job will run 10th on June at 11:15 am

```
15 11 10  JUN SUN echo "Hello cron" >> ~/Desktop/hello.txt
```

Here this job will run on 10th June at 11:15 am if that day is Sunday

```
0,15,30,45  * * * * echo "Hello cron" >> ~/Desktop/hello.txt
```

This job will run every hour in every 15 minutes
0,15,30,45 can be replace with */15

```
*/15  * */3 * * echo "Hello cron" >> ~/Desktop/hello.txt
```

It will run every 15 minutes on 3rd day or every month

```
*/15  * */3  JAN,MAY * echo "Hello cron" >> ~/Desktop/hello.txt
```

It will run every 15 minutes on 3rd day of Jan and May

Executing bash script through CRON job
Bash script backup.sh:

```
#!/bin/bash

tar -czf ~/Desktop/backup.tar.gz ~/{Documents, Downloads, Desktop, Pictures, Videos} 2 >/dev/null
Date >> ~/Desktop/backups.log
```

This is our simple bash file where we are zipping up all the directories inside the curly brace also writing the current date in backups.log file every time this bash script will run. Now we well run this bash using CRON job
crontab -e
To choose the editor

```
15   *  *  *  * bash ~/Desktop/backup.sh
```

This CRON job will run the bash script every 15 minute

BURPSUITE Burp Suite is an integrated platform and graphical tool for performing security testing of web applications, it supports the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.

Gobuster

Gobuster is a software tool for brute forcing directories on web servers

Available Commands:

```
  completion  Generate the auto completion script for the specified shell
  dir         Uses directory/file enumeration mode
  dns         Uses DNS subdomain enumeration mode
  fuzz        Uses fuzzing mode. Replaces the keyword FUZZ in the URL, Headers and the request body
  gcs         Uses gcs bucket enumeration mode
  help        Help about any command
  s3          Uses aws bucket enumeration mode
  tftp        Uses TFTP enumeration mode
  version     shows the current version
  vhost       Uses VHOST enumeration mode (you most probably want to use the IP address as the URL parameter)
```

```
gobuster dns -d google.com -w /usr/share/wordlist/dirduster/directory-list-2.3-medium.txt
```

```
gobuster dir -u 10.10.192.201 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

Will find out directories in the ip address by looking the word file

-u refers url
-w refers the wordlist file to look for

BetterCAP is a powerful, flexible and portable tool created to perform various types of MITM attacks against a network, manipulate HTTP, HTTPS and TCP traffic in real time, sniff for credentials and much more.

> sudo bettercap -iface eth0
Will take you bettercap shell

> help
Here you can find all the modules running/not running

> help net.probe
To get help about specific module e.g net.probe
To run any module type:
net.probe on
net.probe off

> net.show
Wll display as tabular format all the connected clients and their IPs, MAC addresses, vendors etc.

Advanced package tool(APT), is a free-software user interface that works with core libraries to handle the installation and removal of software on Debian, and Debian-based Linux distributions.

apt-get update - Resynchronize sources

apt-get upgrade - Upgrade all installed packages to newest version
apt list to see all the apt package available
--installed option will show only installed package

apt list --upgradable
Show list of package can be upgradable

apt-get dist-upgrade - same as upgrade and also upgrade dependencies apt-cache search keyword/package name

Searching for a Package

Before downloading a software package, you can check whether the package you need is available from your repository, which is a place where your operating system stores information. The apt tool has a search function that can check whether the package is available.

apt-cache show package_name - show information about package

apt-get install - install package

# -y - Auto answer yes

apt-get remove - remove package(but leave configs)

apt-get purge - remove package and config

sudo apt-get autoremove

Will remove those dependencies that were installed with applications an d are no longer used by anything else on the system.

sudo apt-get clean

This will remove any downloaded package file

DPKG – dpkg in Linux is the primary package manager for Debian and Debian-based systems

dpkg -l - Find versions of installed applications

dpkg -i package.deb - Install package

dpkg -r - package.deb

Whoami is an advanced anonymity tool that allows you to stay anonymous on Kali Linux by using +9 powerful privacy modules, including: IP changer (Hides your real IP address) DNS change (Uses privacy-based servers as default DNS servers) Anti-cold boot (Removes system's digital footprint and traces)

Steps:

```
sudo clone https://github.com/omer-dogan/kali-whoami
cd kali-whoami
sudo make install
sudo apt update && sudo apt install tar tor curl python3 python3-scapy network-manager
sudo kali-whoami --start
```

To activate items enter item number then hit enter
Once you seleted and as many item need to be enabled then hit Enter again

```
sudo kali-whoami --stop
```

Base64 is a binary to a text encoding scheme that represents binary data in an American Standard Code for Information Interchange (ASCII) string format. It's designed to carry data stored in binary format across the channels, and it takes any form of data and transforms it into a long string of plain text.

```
echo -n "hello" | openssl base64
```

We are encoding "hello" into base64 ascii string

```
echo aGVsbG8= | base64 -d
```

Here we are decoding the ascii code into human readable format
-d means decode

```
openssl base64 -in hello.txt -out encrypt_hello.txt
```

Here we are encoding a file called hello.txt
-in refers input file
-out refers output file

```
cat encrypt_hello.txt| base64 -d
```

If you want to read the encrypt_hello.txt file.

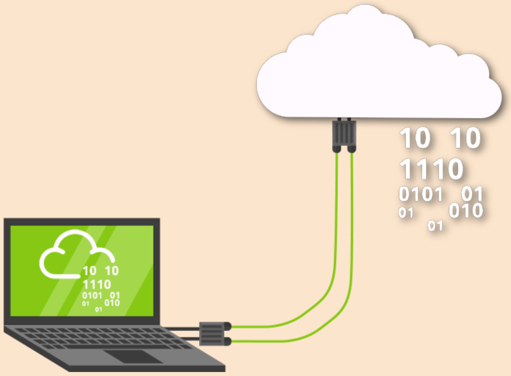
Extra:
ROT13 is a simple letter substitution cipher that replaces a letter with the 13th letter after it in the latin alphabet. ROT13 is a special case of the Caesar cipher which was developed in ancient Rome.

```
rot13
```

Run the command and then enter text to turn into rot13 text
ROT13 is not at all secure. Anyone with a ROT13 decoder can read text encoded with ROT13.

Working with checksum

To verify the official checksum before download any software always check the software release notes of their website



```
ls /usr/bin/*sum
```

To see all available sum as below

```
/usr/bin/b2sum
/usr/bin/innochecksum
/usr/bin/sha1sum
/usr/bin/sha256sum
/usr/bin/sha512sum
/usr/bin/sum
/usr/bin/cksum
/usr/bin/md5sum
/usr/bin/sha224sum
/usr/bin/sha384sum
/usr/bin/shasum
```

samplefile.txt

md5sum samplefile.txt

It will create a md5 hash value as below

```
68201b986072908246fd5ba236588152 samplefile.txt
```

```
sha256 samplefile.txt
```

It will create a sha256 hash value as below

```
105880eccd5694b8217fc6af87a8bb0ffe47ea7453a2b5132bb26468390ce507 samplefile.txt
```

Note:

# It is not a good idea to generate higher value is sha cause it will take long time to generate the hash value.
# If you even add a space in the samplefile.txt file the hash value will change as well

Capturing and Analyzing Packet Packets with TShark and tcpdump

- `apt update; apt install -y tshark`

You will receive a message asking if "non-superusers be able to capture packets". It is good security practice to select no, unless you know what you are doing, or your users need access to the Tshark package.

- `tshark --version`

To check version

Docker

The rest of the scenario will use Docker to simulate a vulnerable host on a network. The gravemind Docker container is obtained from Docker Hub. Multiple hosts will be set up for you using Docker.
Please verify that two gravemind instances are running using the following command:

- docker ps
- tshark
  To start capturing packets
  To stop ctrl+C

TCPDUMP & LiBPCAP Network packet analyzer:

```
man tcpdump
```

# to get help about tcpdump command

```
sudo tcpdump host 192.168.1.10
```

# if you want to listen to or from specific host

```
tcpdump -D
```

# list up all ethernet/network adapters that you have

```
tcpdump -i eth1 -nvXXs0
```

# -i represent interface/source using
# -nv represent no name resolution
-n Don't convert host addresses to names. This can be used to avoid DNS lookups.
# v for verbose
# X represent header information
# X represent ascii information
# s0 represent capturing entire packets

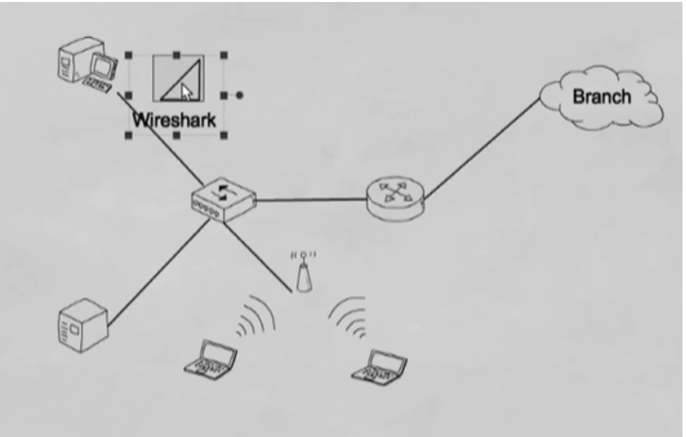
```
tcpdump -i eth1 -nvXXs0 not icmp
# if you want to filter out icmp

tcpdump -i eth1 -nvXXs0 -w tcpDumpTest
# to write the tcpdump result in to file called tcpDumpTest

tcpdump -r tcpDumpTest
# to read the content of the tcpDumpTest file

tcpdump -r tcpDumpTest not icmp
# to read the content of the tcpDumpTest file with filtering
```

Note: You can open any tcpdump output file into Wireshark to visually analyze as well

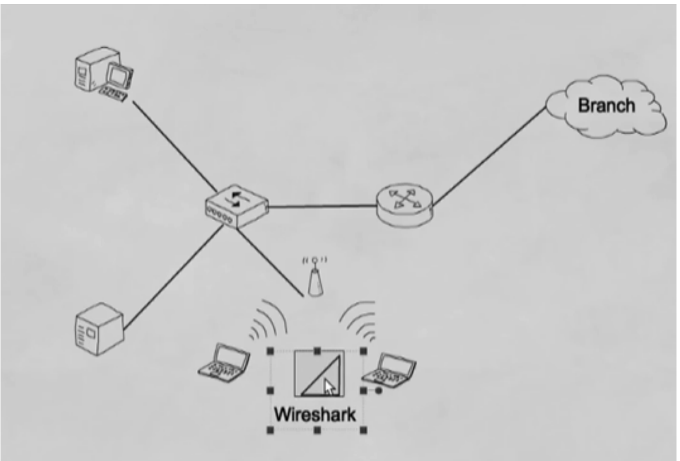
Wireshark is a free, open-source, passive reconnaissance tool that can be used by the Red Team to analyze network traffic. This type of tool is, in general, referred to as a network analyzer, network protocol analyzer, or a packet sniffer. Wireshark can be used in Promiscuous mode which enables it to analyze all the traffic on the network, even if the traffic is not directed to the machine that is running Wireshark, making it possible to view all traffic. The network traffic that is captured can be saved for analysis later. The file format that it is saved as is a .pcapng file format.

Note: You always need to run wireshark as close you can to the actual location where you need to capture network traffic.

If it's wired device:



If it’s wireless divece:



Dumpcap is a network traffic dump tool. It lets you capture packet data from a live network and write the packets to a file. Dumpcap's default capture file format is pcapng format.
Dumpcap is the part of the wireshark suite that captures packets. Unlike Wireshark and tshark, dumpcap cannot see non-physical interfaces like extcap interfaces.

```
dumpcap -D
```

Will display all the interfaces with number

```
dumpcap -i 1 -w /home/output/sample.pcapng -b filesize:10000 -b files:5
```

Here 1 represent the interface number, you can mention eth0 or any other interface by their name
-i refers interface
-w refers the write followed by output directory
-b filesize: 10000 refers the size of the file will be
-b files:5 refers how many files will be created
Note: If you continue capturing packets the files will be overwritten

An in depth look at scanning with Nmap, a powerful network scanning tool.
Network exploration, host discovery, Port scanning, and security auditing.The gui version of nmap is Zenmap.Note: Unauthorized port scanning is prohibited

⚠️Three things to consider while using nmap

1. What type of scan we wanto to do
2. What option do I want go along like, speed
3. Who am I scanning

```
nmap 192.168.4.19
```
Very basic scan

```
nmap  --open 192.168.4.19
```
Only show opon ports or possible open ports

```
nmap -sP 192.168.1.109/24
```
This will san whole network
-s refers service and P refers Ping which is ICMP packets

```
nmap -p- 192.168.1.109  / nmap -p 1-65535 192.168.1.109
```
It will scan all the ports available
-p- refers the port

```
nmap -p20-30 192.169.19.9
```
It will scan between 20 and 30 range

```
nmap -p 80,443,112 192.168.1.109
```
If you want to scan specific ports

```
nmap 192.168.1.109  192.168.1.129
```
To scan multiple IP place them followed by spaces

```
nmap --top-ports 10 192.168.1.109
```
It will return most popular 10 ports on theat IP

```
nmap -oN output.txt 192.168.1.99
```
For write output of the scan
-o for output
```
nmap -oX output.txt 192.168.99
```
For write xml format output

```
nmap -n 192.168.1.99
```

Disable DNS resolution using -n
It will scan much faster without DNS resolution

```
nmap -A 192.168.1.99
```

-A Enable OS detection, version detection, script scanning, and traceroute
Will provide extended information along with the ports

```
nmap -sV 192.168.1.99
```

-sV will provide the version name of the port
-s for service
-V for version

```
nmap -sT 192.168.1.99
```

It will scan only tcp ports
-T refers TCP ports

```
nmap -sU 192.168.1.99
```

It wil lscan UDP ports
-U refers UDP

Steghide is a steganography program that hides data in various kinds of image and audio files , only supports these file formats : JPEG, BMP, WAV and AU. but it's also useful for extracting embedded and encrypted data from other files.

It can be installed with apt however the source can be found on github.

```
steghide embed -ef secret.txt -cf koala.jpg -p khan
```

-ef, --embedfile filename(The file you are embedding to the image)
-cf, --coverfile filename(Specify the cover file that will be used to embed data)
-p refers the password when you will extract the data

```
steghide extract -sf koala.jpg -p khan -xf secretdata.txt
```

-sf --stegofile filename
Specify the stego file (the file that contains embedded data)
-xf, --extractfile filename
Create a file with the name filename and write the data that is embedded in the stego file to it.

Exiftool

Sometimes important stuff is hidden in the metadata of the image or the file , exiftool can be very helpful to view the metadata of the files. You can get it from here

```
exiftool koala.jpg
```

It will display all the metadata information of the image koala.jpg

Wordlists Generators

cewl,crunch,hashcat

cewl - Kali tool

CeWL (Custom Word List generator) is a ruby app which spiders a given URL, up to a specified depth, and returns a list of words which can then be used for password crackers such as John the Ripper. Optionally, CeWL can follow external links.

```
cewl -m 5 -x 6 -d 2 -w khan_custom_words -c -v https://github.com/digininja/CeWL
# Here we are grabbing words from the url given
```

-m 5 means min words length will be 5 chars long
- x 6 means max words length will be 6 chars long
-d 2 means how deep the search will go, there 2 means 2 url down
-c will count each word
-v means verbose
-e means include email address
-w means write followed by filename

crunch - Kali tool

Crunch is a wordlist generator where you can specify a standard character set or any set of characters to be used in generating the wordlists. The wordlists are created through combination and permutation of a set of characters. You can determine the amount of characters and list size.

Create a charset.lst file if you don’t have one with the following content.

```
numeric          = [0123456789]

alpha            = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
alpha-numeric     = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]

loweralpha        = [abcdefghijklmnopqrstuvwxyz]
loweralpha-numeric = [abcdefghijklmnopqrstuvwxyz0123456789]

mixalpha          = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ]
mixalpha-numeric   = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]

# The charset "ascii-32-95" includes all 95 characters on standard US keyboard
ascii-32-95                = [ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuv>
ascii-32-65-123-4           = [ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`{|}~]
alpha-numeric-symbol32-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/ ]
```

```
crunch 7 10 -f charset.lst ascii-32-95 -o bigcrunch
```

7 refers min size word 10 refers max size
-f refers the charset.lst file
-o refers to the output file that will be created by the name bigcrunch
Ascii-32-95 refers one the variable in charset.lst file that contains alpha numeric values
Note: This will generate PB amount of file. So be careful

```
crunch 4 4 -f charset.lst numeric -o pin_crunch
```

# Here we are creating a very small file that will generate 10000 lines of possible pin of 4 digits combination.

```
crunch 7 8 -f charset.lst loweralpha-numeric -o pin_crunch
```

```
crunch 3 3 abcABC123 -o custom_crunch
```

# Here we are creating a combination of 3 alpha numeric chars abcABC123, we are not using any predefined .lst file.

Hashcat

Hashcat is a password recovery tool. It had a proprietary code base until 2015, but was then released as open source software. Versions are available for Linux, macOS, and Windows.

```
echo -n "Hello" | md5sum | cut -d' ' -f1 >> hashes.txt
echo -n "khan" | md5sum| cut -d' ' -f1 >> hashes.txt
echo -n "ahnaf" | md5sum| cut -d' ' -f1 >> hashes.txt
```

The commands above will generate the following hashes accordingly:

```
8b1a9953c4611296a827abf8c47804d7
9e95f6d797987b7da0fb293a760fe57e
A88f72a063223bbb9e1c43d0110e3a80
```

Now we will decode the hashes using hashcat command using a wordlist file

```
hashcat -a 0 -m 0 hashes.txt  /usr/share/wordlist/10_million_password.txt
```

-a means attack mode here is 0
-m means hash type here is 0 which refers md5sum
Each hashing type has specific number assigned to it. i.e sha1 is 100, md4 is 900, sha512 is 1700 etc
Once you hit enter and run the command it will start looking for words in the wordlist file and decode the hashes and along with so many other information you will see the following:

```
8b1a9953c4611296a827abf8c47804d7:Hello
9e95f6d797987b7da0fb293a760fe57e:khan
A88f72a063223bbb9e1c43d0110e3a80:ahnaf
```

Hydra

Hydra is a brute force online password cracking program, a quick system login password “hacking” tool.

```
apt install hydra
Or
dnf install hydra
```

To install Hydra Ubuntu or Fedora Linux system

Brute Forcing SSH

```
hydra -l root -P passwords.txt 192.145.12.10 -t 4 ssh
```

-l specifies the (SSH) username for login / for list of users -L
-P indicates a list of passwords

What if you don’t know the username as well as Password

```
hydra -L  wordlist.txt -P wordlist.txt 192.145.12.10 -t 4 ssh
```

Brute Forcing FTP

```
hydra -l root -P passwords.txt 192.145.12.10 -t 4 ftp
```

Post Web Form

```
hydra -l molly -P rockyou.txt http-post-form “/login:username=^USER^&password=^PASS^:F=incorrect” -V
```

The login page is only /, i.e., the main IP address.

```
hydra -l <username> -P <wordlist> MACHINE_IP http-post-form
"/:username=^USER^&password=^PASS^:F=incorrect" -V
```

rockyou.txt is a password list file
https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

-l refers username
-p indicates the password list
Http-post-form indicates the type of form(POST/GET)
/login indicate the login url
^USER^ specify username(s) will replace ^USER^ (same for ^PASS^ but password will replace)
F=incorrect is a string that appears in the server reply when the login fails
-V verbose output for every attempt

FFUF which is named “Fuzz Faster You Fool” is an open-source web fuzzing tool that discovers elements and content within web applications or web servers in a fast manner. Ffuf has different functionalities Such as fuzz directory, vhost discovery, and Fuzzing based On parameter GET as POST.It's a tool used for web enumeration, fuzzing, and directory brute forcing.

```
ffuf -u https://codingo.io/FUZZ/ -w ./wordlist.txt
```

This is very basic scan
Here ffuf will scan the url https://codingo.io using the wordlist wordlist.txt
-u means target url
-w refers the wordlist
/FUZZ is working like placeholder where the found directories/pages will be matched and retrieved

```
ffuf  -u https://codingo.io/FUZZ -w ./wordlist -recursion -s
```

This is same command like above except it will go under subdirectories/pages
Note: Make sure not to use forward slash after FUZZ
-recursion flag for recursive
-s when you don’t want to see to many buffer infos, only the result

```
ffuf -u https://codingo.io/FUZZ -w ./wordlist -recursion -e .bak
```

This will do everything as before along with find file with specific extensions i.e .bak
-e means extension

```
ffuf -u https://codingo.io/FUZZ/ -w ./wordlist.txt | tee ./output.txt
```

Here we are trying to save the result into the output.txt file using tee command tee command normally used to split the output of a program so that it can be both displayed and saved in a file.

```
ffuf -u https://codingo.io/FUZZ -w ./wordlist.txt -recursion -of html -o output.html
```

This will transfer the output as html file as report
-o refers output
-of refers format you want(csv, json, md, ejson, ecsv)

Example of Username Enumeration:

```
ffuf -w /usr/share/wordlists/SecLists/Usernames/Names/names.txt -X POST -d "username=FUZZ&email=x&password=x&cpassword=x" -H "Content-Type: application/x-www-form-urlencoded" -u http://MACHINE_IP/customers/signup -mr "username already exists"
```

In the above example, the -w argument selects the file's location on the computer that contains the list of usernames that we're going to check exists. The -X argument specifies the request method, this will be a GET request by default, but it is a POST request in our example. The -d argument specifies the data that we are going to send. In our example, we have the fields username, email, password and cpassword. We've set the value of the username to FUZZ. In the ffuf tool, the FUZZ keyword signifies where the contents from our wordlist will be inserted in the request. The -H argument is used for adding additional headers to the request. In this instance, we're setting the Content-Type so the web server knows we are sending form data. The -u argument specifies the URL we are making the request to, and finally, the -mr argument is the text on the page we are looking for to validate we've found a valid username.

Bruteforcing with ffuf:

```
ffuf -w valid_usernames.txt:W1,/usr/share/wordlists/SecLists/Passwords/Common-Credentials/10-million-password-list-top-100.txt:W2 -X POST -d "username=W1&password=W2" -H "Content-Type: application/x-www-form-urlencoded" -u http://MACHINE_IP/customers/login -fc 200
```

This ffuf command is a little different to the previous one in Task 2. Previously we used the FUZZ keyword to select where in the request the data from the wordlists would be inserted, but because we're using multiple wordlists, we have to specify our own FUZZ keyword. In this instance, we've chosen W1 for our list of valid usernames and W2 for the list of passwords we will try. The multiple wordlists are again specified with the -w argument but separated with a comma. For a positive match, we're using the -fc argument to check for an HTTP status code other than 200.

John the Ripper is a popular password cracking tool that can be used to perform brute-force attacks using different encryption technologies and helpful wordlists. It's often what pen-testers and ethical hackers use to find the true passwords behind hashes.

Example of cracking a password protected zip file:

```
# Let's create a text file with some content in it
nano secret.txt

# Compress the file with zip along with password
zip -P khan2007 private.zip secret.txt

# Collecting hashing information and saving to tobehacked.txt file
zip2john private.zip > tobehacked.txt

# This will run the password cracking process along with verbose
john tobehacked.txt

# Just to see the extracted password
john tobehacked.txt --show
```

Example of cracking Linux user password:

```
unshadow /etc/passwd /etc/shadow | tail -n 6 > hashes
```

Here we are creating hashes file from unshadow passwd and shadow file and grabbing only last 6 lines
One of them is called “unshadow”. The unshadow command combines the passwd and shadow files together into a single file. This can then be used by John to crack passwords.

```
john -wordlist:/home/kali/pass.lst --format=crypt hashes
```

Here we are cracking the password from hashes file created earlier and using the pass.lst wordlist file that was also created.
Note: If you use --wordlist then use equal sign instead of colon

```
john --show hashes
```

If you run this command after the cracking using earlier command it will just show the result only without verbose

```
john -wordlist:/home/kali/pass.lst -rules:AppendJustNumbers --format=crypt hashes
```

Here we are adding rules