Introduction Linux

>> Firefox ESR (web browser),

>> Terminal (command prompt),

>> Metasploit framework,

>> Burp suite (use for web application penetration testing),

>> Zen map (tool) graphical version tool of NMap,

>> Cherry Tree (taking notes we use this),

Default browser link: in VMware browser: file:///usr/share/kali-defaults/web/homepage.html

Navigating the file system

List of commands we use in Linux terminal:

>> **pwd** (present working directory)

>> **cd..** (Go backwards)

>> **ls** (list - everything in the folder)

>> **sudo su** (to get access root) // here password is required for root access

>> **cd** / (folder name)/ // (go forwards)

>> **~ atilda** (shows the home folder)

Example: **~/Desktop/** or **~/Music/** (we can directly switch folder if we use this)

>> **cd D** (it will show the existing files which you can start with D)

>> **mkdir folder_name** (Creates folder in path)

>> **rmdir folder_name** (Deletes folder in path)

>> **ls -la** (Shows the hidden files and folder)

>> **echo "Text"** > filename.txt (echo write and > create file.txt in required path)

>> **cp filename.txt foldername/** (we can "cp" copy file to required folder)

>> **mv filename.txt foldername/** (we can "mv" move file to required folder)

>> **rm foldername/filename.txt** (this will remove rm filename.txt from folder)

>> **locate bash** (this will show locate shell like bash) if not we try to update with updatedb

>> **man pages** (that is instruction pages for any command that you are running)

-> **man ls**

-> **ls --help**

Users and privileges

Notations and explanations

>> **ls -la** (shows hidden files in list)

- >> file

d >> directory

r >> read

w >> write

x >> execute

There are 3 groups: Example (rwx xrw -rx)

1st group: OWNER (is the owner of the file)

2nd group: GROUP (permission of the member of the group who owns file, group ownership)

3rd group: PUBLIC

- -

Create file and print

ls -la /tmp/ // in pen testing upload rather exploit must be on tmp folder, and here we can execute those files.

echo "text" > filename.txt (this will write text inside and create file with hello folder)

cat filename.txt (this will print file something written in it)

- -

Allocate the permission

1st way

chmod +rwx

2nd way (number feature)

>>**chmod 777 filename.txt** // full permission -rwxrwxrwx

>>**ls -ltr** (shows the file permission in specific folders)

- -

Add New User

>>**adduser username** (add the user)

>> **cat /etc/user** (this will print all the user)

>> **cat /etc/shadow** (it is in hashing format, break down by tool using by hashcat breakdown password)

>> **su username** (switch user "type username")

>> **passwd root** (change the password for user)

Common networking commands

>> ifconfig It shows different interface types and ip associated with it

inet

subnet

broadcast

MAC address

also have loop back address::

/// if your machine has a wireless adaptor or do wireless penetration testing we "**iwconfig**"

>> **arp -a** (it shows the ip address who talk to mac address associated with it)

>> **netstat -ano** (it shows the active connection that you are running machine)

>> **route** (print routing table and tell where your traffic access) \\ if more than one network is available or talking to both or switching one to the other in same machine called piviting

View create edit in files

- Create file

>> **echo filename.txt** (create file) ex: echo hello

>> **cat** (print the content of file)

>> **">"** (append text or add text to next line)

>> **touch filename** (create the blank file)

>> **nano filename.txt** (this graphical way to create new file)

- open text editor in terminal
-

>> **nano filename.txt** (open editor you can type the anything) \\ use - create scripts/python scripts/Edit shell code/ in Exploit development

>> **gedit filename.txt** (same you can open and type whatever you want)

Starting and Stopping Kali Services

>> **service apache2 start** (run apache webserver on local computer with this command) \\ also create malicious webpage

(Spin up web server)

-m is module

80 is port

>> **python -m SimpleHTTPerver 8080** (start server and accessing files in it)

>> **systemctl enable postgresql** (systemctl - this will create and keep the system entire time// also it will allow us to run Metasploit and PostgreSQL running)

Installing and Updating Tools

First method

>> **apt update && apt upgrade** (this will update and upgrade system if we press yes)

Second method

This is the website: https://unix.stackexchange.com/questions/525106/unable-to-locate-package-python-pip

Edit your `sources.list` (as root):

0

```
nano /etc/apt/sources.list
```

with the following lines:

```
deb http://deb.debian.org/debian stretch main
deb-src http://deb.debian.org/debian stretch main

deb http://deb.debian.org/debian-security/ stretch/updates main
deb-src http://deb.debian.org/debian-security/ stretch/updates main

deb http://deb.debian.org/debian stretch-updates main
deb-src http://deb.debian.org/debian stretch-updates main
```

Ctrl + O , Enter , Ctrl + X to save , then run:

```
apt update
apt install python-pip
```

Example sources.list

Package: python-pip

List of command I have done manually install python in kali Linux

>> This is the website: https://www.kalilinux.in/2019/08/install-python3-in-kali-linux.html

>> **apt-get update**

>> **cd Downloads**

>> **tar -xvf Python-3.x.x.tar.xz**

>> **cd Python3.x.x**

>> **. /configure**

>> **make**

>> **make install**

>> **python3 -V** (if not update after follow above procedure)

>> **apt-get install python3-pip**

Third Method

GitHub clone search on google: GitHub impacket

Link copied https://github.com/SecureAuthCorp/impacket.git

>> **cd /opt/**

>> **git clone https://github.com/SecureAuthCorp/impacket.git**

>> **cd impacket/**

>> **pip install**. (It will download the entire file we need)

Scripting with Bash

Building out ping swipper script

Commands

>> **grep** (specified data would narrow down results) Example: grep "specify the line"

>> **cut** (cut down the scripts)

>> **tr** (translating or deleting characters)

>> **-c** (count 1) Example: -c -1

>> **|** (pipe: allow writing additional command)

>> **cut -d " " -f 4** (-d means delimiter what we are cutting on " " << this is)

(-f means field: what field we want to retrieve it will retrieve 4th number of field)

>> **&** (use for threading if we not use this then only single ip it will take and run)

>> **$** (we taking input from user)

>> **-sS** (steal Scan)

>> **-p** (port)

>> **-T4** (speed)

Writing Script

>> ping 192.168.182.128

>> ping 192.168.182.128 -c 1

```
root@kali:/home/kali# ping 192.168.182.128 -c 1
PING 192.168.182.128 (192.168.182.128) 56(84) bytes of data.
64 bytes from 192.168.182.128: icmp_seq=1 ttl=64 time=0.028 ms
```

>> ping 192.169.182.128 -c 1 > ip.txt (create the file with > extension)

>> cat ip.txt

```
root@kali:/home/kali# cat ip.txt
PING 192.168.182.128 (192.168.182.128) 56(84) bytes of data.
64 bytes from 192.168.182.128: icmp_seq=1 ttl=64 time=0.030 ms
64 bytes from 192.168.182.128: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 192.168.182.128: icmp_seq=3 ttl=64 time=0.059 ms
64 bytes from 192.168.182.128: icmp_seq=4 ttl=64 time=0.060 ms
```

>> cat ip.txt | grep "64 bytes"

```
root@kali:/home/kali# cat ip.txt | grep "64 bytes"
64 bytes from 192.168.182.128: icmp_seq=1 ttl=64 time=0.069 ms
```

>> cat ip.txt | grep "64 bytes" | cut -d " " -f 4

```
root@kali:/home/kali# cat ip.txt | grep "64 bytes" | cut -d " " -f 4
192.168.182.128:
```

>> cat ip.txt | grep "64 bytes" | cut -d " " -f 4 | tr -d ":"

```
root@kali:/home/kali# cat ip.txt | grep "64 bytes" | cut -d " " -f 4 | tr -d ":"
192.168.182.128
```

Now we open EDITOR (leaf pad) and create file **ipshweep.sh**

#!/bin/bash

for ip in $(seq 1 254); do

ping -c 1 $1.$ip | grep "64bytes" | cut -d " " -f 4 | tr -d ":" &

done

>> **chmod +x ipshweep.sh** (executable changing permissions)

>> **./ipshweep.sh** (because we don't have provide any ip address)

```
root@kali:/home/kali# ./ipshweep.sh
ping: .1: Name or service not known
ping: .2: Name or service not known
ping: ping: .3: Name or service not known
.7: Name or service not known
```

>> **./ipshweep.sh 192.168.1** (this will pull list of ip)

```
root@kali:~# ./ipsweep.sh 192.168.1
192.168.1.90
192.168.1.74
192.168.1.73
192.168.1.79
192.168.1.80
192.168.1.77
192.168.1.65
```

>> **./ipshweep.sh 192.168.1 > iplist.txt** (create the file)

changes in script

```
#!/bin/bash
if [ "$1" == "" ]
then
echo "You forgot an IP address!"
echo "Syntax: ./ipsweep.sh 192.168.1"

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

>> **./ipsweep.sh** (run the script with)

Looping (suppose if we have list of ip and we have to do scan all ip)

>> **for ip in $(cat iplist.txt); do nmap -sS -p 80 -T4; done // for single ip test**

>> **for ip in $(cat iplist.txt); do nmap -sS -p 80 -T4 $ip & done // for multiple ip test**

```
root@kali:/home/kali# for ip in $(cat iplist.txt); do nmap -sS -p 80 -T4 $ip & done
[1] 12376
[2] 12377
[3] 12378
[4] 12379
[5] 12380
[6] 12381
[7] 12382
root@kali:/home/kali# Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 07:43 EDT
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 07:43 EDT
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 07:43 EDT
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 07:43 EDT
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 07:43 EDT
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-10 07:43 EDT
Nmap scan report for 192.168.1.73
Host is up (0.0040s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap done: 1 IP address (1 host up) scanned in 13.59 seconds
Nmap scan report for 192.168.1.77
Host is up (0.00065s latency).

PORT   STATE    SERVICE
80/tcp filtered http

Nmap done: 1 IP address (1 host up) scanned in 13.60 seconds
Nmap scan report for 192.168.1.90
```