1. Use the ping command to test the connectivity to a remote server (e.g., example.com).

```
┌──(surya_jjp㉿kali)-[~]
└─$ ping -c 4 example.com
PING example.com (93.184.215.14) 56(84) bytes of data.
64 bytes from 93.184.215.14: icmp_seq=1 ttl=49 time=595 ms
64 bytes from 93.184.215.14: icmp_seq=2 ttl=49 time=311 ms
64 bytes from 93.184.215.14: icmp_seq=3 ttl=49 time=334 ms
64 bytes from 93.184.215.14: icmp_seq=4 ttl=49 time=309 ms

── example.com ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 309.152/387.225/594.590/120.110 ms
```

2. Write a script to measure the round-trip time for each packet and analyze the results.

```
┌──(surya_jjp㉿suryajjp)-[~]
└─$ cat rtt.sh
#!/bin/bash

# Server to ping (you can change this to any other server)
SERVER="google.com"

# Ping the server and extract the round-trip time
echo "Pinging $SERVER ..."
ping -c 10 $SERVER | grep "time=" | awk -F"time=" '{print $2}' | awk '{print $1}' > rtt_times
.txt

# Check if any RTT values were captured
if [[ ! -s rtt_times.txt ]]; then
    echo "No round-trip times recorded. Please check the network or DNS resolution."
    exit 1
fi

# Analyze the round-trip times
echo "Round-trip times (RTT) in ms:"
cat rtt_times.txt

# Calculate the average RTT
avg_rtt=$(awk '{sum+=$1} END {print sum/NR}' rtt_times.txt)
echo "Average RTT: $avg_rtt ms"

# Calculate the minimum RTT
min_rtt=$(awk 'NR == 1 {min = $1} {if ($1 < min) min = $1} END {print min}' rtt_times.txt)
echo "Minimum RTT: $min_rtt ms"

# Calculate the maximum RTT
max_rtt=$(awk 'NR == 1 {max = $1} {if ($1 > max) max = $1} END {print max}' rtt_times.txt)
echo "Maximum RTT: $max_rtt ms"

# Count the number of packets received (successful pings)
packet_count=$(wc -l < rtt_times.txt)
echo "Total packets received: $packet_count"

# Clean up
rm -f rtt_times.txt
```

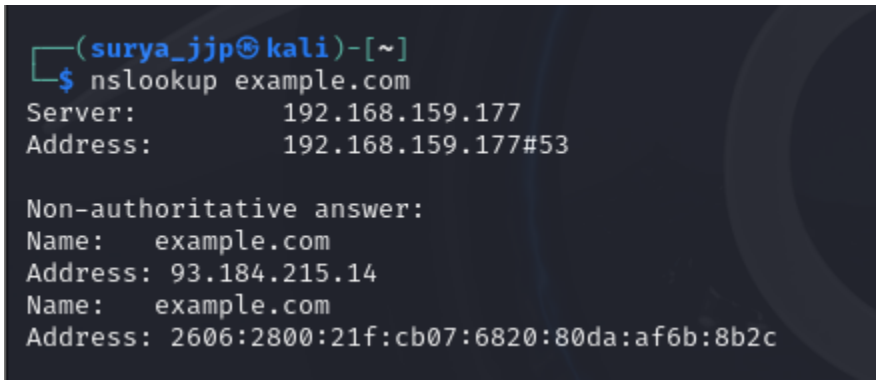
3. Use the traceroute to trace the route packets take to a destination

```
┌──(surya_jjp㉿kali)-[~]
└─$ traceroute example.com
traceroute to example.com (93.184.215.14), 30 hops max, 60 byte packets
 1  192.168.159.177 (192.168.159.177)  3.573 ms  4.780 ms  4.945 ms
 2  * * *
 3  10.40.12.141 (10.40.12.141)  78.255 ms  78.203 ms  77.945 ms
 4  10.40.9.93 (10.40.9.93)  69.970 ms  70.090 ms  69.983 ms
 5  aes-static-073.44.22.125.airtel.in (125.22.44.73)  77.726 ms  77.320 ms  77.225 ms
 6  116.119.112.132 (116.119.112.132)  178.134 ms 116.119.73.117 (116.119.73.117)  148.855 ms 116.119.57.82 (116.119
.57.82)  160.127 ms
 7  mei-b5-link.ip.twelve99.net (62.115.42.118)  167.324 ms  167.306 ms  160.037 ms
 8  prs-bb1-link.ip.twelve99.net (62.115.124.54)  189.470 ms prs-bb2-link.ip.twelve99.net (62.115.124.56)  189.460 m
s prs-bb1-link.ip.twelve99.net (62.115.124.54)  177.171 ms
 9  * ash-bb2-link.ip.twelve99.net (62.115.112.242)  295.454 ms  295.442 ms
10  ash-b2-link.ip.twelve99.net (62.115.123.125)  289.542 ms  262.795 ms ash-b2-link.ip.twelve99.net (62.115.123.123
)  262.038 ms
11  62.115.175.71 (62.115.175.71)  270.752 ms  264.920 ms  272.812 ms
12  ae-65.core1.dcd.edgecastcdn.net (152.195.64.153)  278.422 ms  268.555 ms ae-66.core1.dcd.edgecastcdn.net (152.19
5.65.153)  283.216 ms
13  93.184.215.14 (93.184.215.14)  262.284 ms  262.911 ms  262.900 ms
14  93.184.215.14 (93.184.215.14)  262.771 ms  259.197 ms  265.947 ms
```

4. Analyze the output to identify any potential bottlenecks or points of failure in the route.

Hop 2 shows no response(* * *). The reason could be either due to packet loss or firewall filtering blocking ICMP (ping) packets at this hop

5. Use the nslookup command to find the IP address of a given domain (e.g., example.com).



6. Use the netstat command to view active connections and listening ports on your machine.

```
┌──(surya_jjp㉿kali)-[~]
└─$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State

┌──(surya_jjp㉿kali)-[~]
└─$ netstat -tune
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User       Inode
udp        0      0 192.168.159.1:68        192.168.159.177:67      ESTABLISHED 0          113412

┌──(surya_jjp㉿kali)-[~]
└─$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 192.168.159.1:bootpc    192.168.159.177:bootps  ESTABLISHED
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     8542
unix  3      [ ]         STREAM     CONNECTED     8038     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9159     @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     8796     /run/user/1001/at-spi/bus_0
unix  3      [ ]         DGRAM      CONNECTED     5472
```

7. Use the ifconfig (Linux) or ip a command to display network interface configurations.

```
┌──(surya_jjp㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.159.1  netmask 255.255.255.0  broadcast 192.168.159.255
        inet6 fe80::a00:27ff:fedc:2cca  prefixlen 64  scopeid 0x20<link>
        inet6 2401:4900:16ed:b79f:559f:ec70:c62d:9446  prefixlen 64  scopeid 0x0<global>
        inet6 2401:4900:16ed:b79f:a00:27ff:fedc:2cca  prefixlen 64  scopeid 0x0<global>
        ether 08:00:27:dc:2c:ca  txqueuelen 1000  (Ethernet)
        RX packets 4443  bytes 348028 (339.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1058  bytes 121635 (118.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 24  bytes 1616 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1616 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

8. Write a script to report document the configuration of each interface, noting the IP address, subnet mask, and any other relevant information.

```
┌──(surya_jjp㉿kali)-[~]
└─$ ip -o -4 addr show | awk '{print "Interface: "$2"\nIP Address: "$4"\n"}'
Interface: lo
IP Address: 127.0.0.1/8

Interface: eth0
IP Address: 192.168.159.1/24
```

9. Perform a basic network scan using nmap on your local network to identify active devices and open ports.

```
┌──(surya_jjp㉿kali)-[~]
└─$ nmap 192.168.159.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-19 18:23 IST
Nmap scan report for 192.168.159.1
Host is up (0.00021s latency).
All 1000 scanned ports on 192.168.159.1 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.159.177
Host is up (0.012s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
53/tcp open  domain

Nmap done: 256 IP addresses (2 hosts up) scanned in 2.85 seconds
```

10. Create a report summarizing the devices found, their IP addresses, and the services running on the open ports.

```
┌──(surya_jjp㉿kali)-[~]
└─$ nmap -sP 192.168.159.0/24 -oN devices_report.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-19 18:40 IST
Nmap scan report for 192.168.159.1
Host is up (0.00063s latency).
Nmap scan report for 192.168.159.177
Host is up (0.0038s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 2.62 seconds

┌──(surya_jjp㉿kali)-[~]
└─$ cat de
desktop/           devices_report.txt
┌──(surya_jjp㉿kali)-[~]
└─$ cat devices_report.txt
# Nmap 7.94SVN scan initiated Sat Oct 19 18:40:07 2024 as: nmap -sP -oN devices_report.txt 192.168.159.0/24
Nmap scan report for 192.168.159.1
Host is up (0.00063s latency).
Nmap scan report for 192.168.159.177
Host is up (0.0038s latency).
# Nmap done at Sat Oct 19 18:40:09 2024 -- 256 IP addresses (2 hosts up) scanned in 2.62 seconds
```

11. Capture network packets using tcpdump on a specific interface.

```
┌──(kali㉿suryajjp)-[~]
└─$ sudo tcpdump -i eth0 -w capture.pcap
sudo: unable to resolve host suryajjp: Name or service not known
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C24 packets captured
24 packets received by filter
0 packets dropped by kernel
```

12. Analyze the captured packets for specific protocols (like HTTP or DNS) and summarize your findings.

13. Use the whois command to gather registration information about a domain.

```
┌──(surya_jjp㉿kali)-[~]
└─$ whois example.com
   Domain Name: EXAMPLE.COM
   Registry Domain ID: 2336799_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.iana.org
   Registrar URL: http://res-dom.iana.org
   Updated Date: 2024-08-14T07:01:34Z
   Creation Date: 1995-08-14T04:00:00Z
   Registry Expiry Date: 2025-08-13T04:00:00Z
   Registrar: RESERVED-Internet Assigned Numbers Authority
   Registrar IANA ID: 376
   Registrar Abuse Contact Email:
   Registrar Abuse Contact Phone:
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Name Server: A.IANA-SERVERS.NET
```

14. Use the hostname command to display and change the hostname of your machine.

```
┌──(surya_jjp㉿kali)-[~]
└─$ hostname
kali
```

```
┌──(kali㉿kali)-[~]
└─$ sudo hostnamectl set-hostname surya_jjp

┌──(kali㉿kali)-[~]
└─$ hostname
suryajjp
```

15. Use the finger command to gather information about users on a system.

```
┌──(kali㉿kali)-[~]
└─$ sudo finger
Login       Name       Tty      Idle  Login Time   Office     Office Phone
surya_jjp              tty7       18  Oct 20 11:15 (:0)
```

16. Use the who command to see who is currently logged into the system and the last command to view the login history.

```
┌──(kali㉿kali)-[~]
└─$ who
surya_jjp tty7         Oct 20 11:15 (:0)
```

```
┌──(kali㉿kali)-[~]
└─$ last
surya_jj tty7         :0               Sun Oct 20 11:15   still logged in
reboot   system boot  6.6.15-amd64     Sun Oct 20 11:14   still running
surya_jj tty7         :0               Fri Oct 18 14:49 - crash (1+20:25)
reboot   system boot  6.6.15-amd64     Fri Oct 18 14:48   still running
surya_jj tty7         :0               Wed Oct 16 11:46 - crash (2+03:01)
reboot   system boot  6.6.15-amd64     Wed Oct 16 11:45   still running
surya_jj tty7         :0               Tue Oct 15 16:09 - crash  (19:36)
reboot   system boot  6.6.15-amd64     Tue Oct 15 15:59   still running
surya_jj tty7         :0               Sat Oct 12 18:43 - crash (2+21:15)
reboot   system boot  6.6.15-amd64     Sat Oct 12 18:41   still running
kali     tty7         :0               Wed Oct  9 14:19 - crash (3+04:21)
reboot   system boot  6.6.15-amd64     Wed Oct  9 14:17   still running
```

Xargs

---

1. Write a shell script called testurl.sh that accepts a list of urls in a separate file and tests if the website is up or not.

```
┌──(surya_jjp㉿suryajjp)-[~]
└─$ cat > testurl.sh
#!/bin/bash
# Usage: ./testurl.sh urls.txt
cat $1 | xargs -n 1 -I {} bash -c '
if curl -s --head --request GET {} | grep "200 OK" > /dev/null; then
    echo "{} is up"
else
    echo "{} is down"
fi'
```

```
┌──(surya_jjp㉿suryajjp)-[~]
└─$ cat > urls.txt
https://www.google.com
https://www.nonexistentwebsite123.com
https://www.github.com
```

```
┌──(surya_jjp㉿suryajjp)-[~]
└─$ ./testurl.sh urls.txt
xargs: warning: options --max-args and --replace/-I/-i are mu
ous --max-args value
https://www.google.com is down
https://www.nonexistentwebsite123.com is down
https://www.github.com is down
```

2. Create a shell script called diskhog.sh that lists the 5 largest items (files or directories) in the current directory in decreasing order of size.

```
┌──(surya_jjp㉿suryajjp)-[~]
└─$ du -ah . | sort -rh | head -n 5
23M     .
12M     ./.mozilla/firefox/z9bce81y.default-esr
12M     ./.mozilla/firefox
12M     ./.mozilla
9.5M    ./.cache
```

3. compress all .log files found in the /var/logs/ directory?

```
┌──(kali㉿suryajjp)-[~]
└─$ cat > log.sh
#!/bin/bash
find /var/logs/ -type f -name "*.log" | xargs gzip

┌──(kali㉿suryajjp)-[~]
└─$ ./log.sh
zsh: permission denied: ./log.sh

┌──(kali㉿suryajjp)-[~]
└─$ chmod +x log.sh

┌──(kali㉿suryajjp)-[~]
└─$ ./log.sh
find: '/var/logs/': No such file or directory
gzip: compressed data not written to a terminal. Use -f to force compression.
For help, type: gzip -h
```

4. delete all temporary files older than 7 days from the /tmp/ directory?

```
┌──(kali㉿suryajjp)-[~]
└─$ cat > sample.sh
#!/bin/bash
find /tmp/ -type f -mtime +7 | xargs rm
```

5. write a shell script to make all .sh files in your home directory executable?

```
┌──(kali㉿suryajjp)-[~]
└─$ cat sample.sh
#!/bin/bash
find ~ -type f -name "*.sh" | xargs chmod +x

┌──(kali㉿suryajjp)-[~]
└─$ ls -l | grep ".*sh$"
-rwxrwxr-x 1 kali    kali         97 Aug  8 11:43 hobby.sh
-rwxrwxr-x 1 kali    kali        735 Oct  8 23:06 lab5.sh
-rwxrwxr-x 1 kali    kali         63 Oct 20 12:53 log.sh
-rwxrwxr-x 1 kali    kali        197 Sep  3 01:07 rectify.sh
-rwxrwxr-x 1 kali    kali         57 Oct 20 12:56 sample.sh
-rwxrwxr-x 1 kali    kali        347 Oct  8 23:40 script.sh
```

6. search for the string "auth" in all .conf files in the /etc/ directory

```
┌──(kali㉿suryajjp)-[~]
└─$     cat > sample.sh
#!/bin/bash
find /etc/ -type f -name "*.conf" | xargs grep -i "auth"

┌──(kali㉿suryajjp)-[~]
└─$ ./sample.sh
find: '/etc/redis': Permission denied
find: '/etc/vpnc': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/credstore.encrypted': Permission denied
find: '/etc/credstore': Permission denied
find: '/etc/openvas/gnupg': Permission denied
find: '/etc/ipsec.d/private': Permission denied
find: '/etc/polkit-1/rules.d': Permission denied
/etc/security/faillock.conf:# authentication attempts.
/etc/security/faillock.conf:# Only track failed user authentications attempts f
/etc/security/faillock.conf:# authentication attempts. Enabling this option wil
```

7. count the number of "failed" login attempts in all .log files in /var/log/?

```
┌──(kali㉿suryajjp)-[~]
└─$ cat > sample.sh
#!/bin/bash
find /var/log/ -type f -name "*.log" | xargs grep -c "failed"

┌──(kali㉿suryajjp)-[~]
└─$ ./sample.sh
find: '/var/log/inetsim': Permission denied
find: '/var/log/private': Permission denied
find: '/var/log/speech-dispatcher': Permission denied
find: '/var/log/lightdm': Permission denied
/var/log/macchanger.log:0
/var/log/Xorg.0.log:201
/var/log/Xorg.2.log:0
/var/log/nginx/access.log:0
/var/log/nginx/error.log:0
/var/log/apache2/access.log:0
/var/log/apache2/error.log:0
/var/log/apache2/other_vhosts_access.log:0
/var/log/fontconfig.log:0
/var/log/apt/term.log:0
/var/log/apt/history.log:0
/var/log/dpkg.log:0
/var/log/stunnel4/stunnel.log:0
/var/log/postgresql/postgresql-16-main.log:0
/var/log/alternatives.log:0
grep: /var/log/boot.log: Permission denied
/var/log/Xorg.1.log:0
```

8. rename all .txt files in the current directory by appending .bak

```
┌──(kali㉿suryajjp)-[~]
└─$ cat > sample.sh
#!/bin/bash
find . -type f -name "*.txt" | xargs -I {} mv {} {}.bak

┌──(kali㉿suryajjp)-[~]
└─$ ./sample.sh

┌──(kali㉿suryajjp)-[~]
└─$ ls | grep "*.txt"

┌──(kali㉿suryajjp)-[~]
└─$ ls | grep "*.bak"

┌──(kali㉿suryajjp)-[~]
└─$ ls -l | grep ".bak$"
-rw-rw-r-- 1 kali    kali    3293613 Sep 12 12:12 English.txt.bak
-rw-rw-r-- 1 kali    kali         11 Aug 31 00:30 a1.txt.bak
-rw-rw-r-- 1 kali    kali       2399 Oct  8 23:46 contents-sorted.txt.bak
-rw-rw-r-- 1 kali    kali       2399 Oct  8 23:44 contents.txt.bak
-rw-rw-r-- 1 kali    kali         13 Oct  3 11:17 encrypted_file.txt.bak
-rw-rw-r-- 1 kali    kali          2 Sep 25 15:22 example.txt.bak
-rw-rw-r-- 1 kali    kali         28 Oct  8 23:54 field2.txt.bak
-rw-rw-r-- 1 kali    kali         13 Oct  3 10:25 file.txt.bak
-rw-rw-r-- 1 kali    kali         41 Aug  8 11:34 hobby.txt.bak
-rw-rw-r-- 1 kali    kali         81 Aug 31 12:32 input.txt.bak
-rw-rw-r-- 1 kali    kali          0 Aug  8 00:00 newfile.txt.bak
-rw-rw-r-- 1 kali    kali          0 Aug 31 11:45 payload.txt.bak
-rw-rw-r-- 1 kali    kali          0 Aug 25 22:52 sample.txt.bak
```

9. Write a shell script to check if a list of users from users.txt exist in the system.

```
┌──(kali㉿suryajjp)-[~]
└─$ cat > sample.sh
#!/bin/bash
cat users.txt | xargs -I {} bash -c '
if id -u {} >/dev/null 2>&1; then
    echo "{} exists"
else
    echo "{} does not exist"
fi'
```

10. search for keywords like "ERROR" or "CRITICAL" in all log files over 1MB in size.

```
┌──(kali㉿suryajjp)-[~]
└─$ cat > sample.sh
#!/bin/bash
find /var/log/ -type f -size +1M -name "*.log" | xargs grep -Ei "ERROR|CRITICAL"
```

==========

==========

“If everyone is moving forward together, then success takes care of itself.” — Henry Ford