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

```
└─$ ping 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=128 time=230 ms
64 bytes from 93.184.215.14: icmp_seq=2 ttl=128 time=230 ms
64 bytes from 93.184.215.14: icmp_seq=3 ttl=128 time=236 ms
64 bytes from 93.184.215.14: icmp_seq=4 ttl=128 time=238 ms
64 bytes from 93.184.215.14: icmp_seq=5 ttl=128 time=233 ms
64 bytes from 93.184.215.14: icmp_seq=6 ttl=128 time=229 ms
64 bytes from 93.184.215.14: icmp_seq=7 ttl=128 time=251 ms
64 bytes from 93.184.215.14: icmp_seq=8 ttl=128 time=233 ms
64 bytes from 93.184.215.14: icmp_seq=9 ttl=128 time=230 ms
64 bytes from 93.184.215.14: icmp_seq=10 ttl=128 time=238 ms
64 bytes from 93.184.215.14: icmp_seq=11 ttl=128 time=252 ms
64 bytes from 93.184.215.14: icmp_seq=12 ttl=128 time=255 ms
64 bytes from 93.184.215.14: icmp_seq=13 ttl=128 time=250 ms
64 bytes from 93.184.215.14: icmp_seq=14 ttl=128 time=231 ms
64 bytes from 93.184.215.14: icmp_seq=15 ttl=128 time=247 ms
64 bytes from 93.184.215.14: icmp_seq=16 ttl=128 time=253 ms
64 bytes from 93.184.215.14: icmp_seq=18 ttl=128 time=234 ms
```

```
64 bytes from 93.184.215.14: icmp_seq=45 ttl=128 time=230 ms
64 bytes from 93.184.215.14: icmp_seq=46 ttl=128 time=233 ms
^C
── example.com ping statistics ──
46 packets transmitted, 45 received, 2.17391% packet loss, time 45072ms
rtt min/avg/max/mdev = 228.684/236.077/255.493/7.857 ms
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ ./lab_8_2.sh

Min RTT: max ms, Avg RTT: 232.984 ms, Max RTT: 5.058 ms ms
```

```
#!/bin/bash

# Set the target host
TARGET="example.com"

# Ping the target and extract round-trip times
ping -c 10 $TARGET | awk -F'/' 'END {print "Min RTT: " $3 " ms, Avg RTT: " $5 " ms, Max RTT: " $7 " ms"}'
```

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

```
└─$ traceroute example.com
traceroute to example.com (93.184.215.14), 30 hops max, 60 byte packets
 1  192.168.68.2 (192.168.68.2)  0.788 ms  0.707 ms  0.654 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

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

Analyze Traceroute Output

When analyzing the output of the `traceroute`, look for:

- **High Latency**: Identify any hops with significantly higher response times.
- **Timeouts**: Any `* * *` entries indicate that a hop did not respond. This may suggest a potential bottleneck or a firewall blocking ICMP packets.
- **Consistent Delays**: If a hop consistently shows delays, it could be a point of failure.

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ nslookup example.com
Server:         192.168.68.2
Address:        192.168.68.2#53

Non-authoritative answer:
Name:   example.com
Address: 93.184.215.14
Name:   example.com
Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c
```

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

his command shows:

- `-t`: TCP connections
- `-u`: UDP connections
- `-l`: only listening ports
- `-n`: show numerical addresses instead of resolving hostnames

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     7700     /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     8974     /tmp/ssh-WpQptTdqMY70/agent.1072
unix  2      [ ACC ]     STREAM     LISTENING     8163     /tmp/.ICE-unix/1072
unix  2      [ ACC ]     STREAM     LISTENING     406      /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     408      /run/pcscd/pcscd.comm
unix  2      [ ACC ]     STREAM     LISTENING     7981     /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     7990     /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     7991     /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     7993     /run/user/1000/gcr/ssh
unix  2      [ ACC ]     STREAM     LISTENING     7995     /run/user/1000/keyring/control
unix  2      [ ACC ]     STREAM     LISTENING     7997     /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     7999     /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     8001     /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     8003     /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     8005     /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     3691     /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     8007     /run/user/1000/pipewire-0
unix  2      [ ACC ]     STREAM     LISTENING     8009     /run/user/1000/pipewire-0-manager
unix  2      [ ACC ]     STREAM     LISTENING     3693     /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     3694     /run/systemd/io.systemd.ManagedOOM
unix  2      [ ACC ]     STREAM     LISTENING     6660     /run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     6665     /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     6667     /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     8947     /run/user/1000/keyring/pkcs11
unix  2      [ ACC ]     STREAM     LISTENING     9010     /run/user/1000/at-spi/bus_0
unix  2      [ ACC ]     STREAM     LISTENING     7092     /run/systemd/journal/io.systemd.journal
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:b8:ed:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.68.132/24 brd 192.168.68.255 scope global dynamic noprefixroute eth0
       valid_lft 1339sec preferred_lft 1339sec
    inet6 fe80::7cf1:1bd7:5c1c:973e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ ./lab_8_8.sh
Interface: eth0
IP Address: 192.168.68.132
Subnet Mask: 24
MAC Address: 00:0c:29:b8:ed:0d
----------------------------------------
Interface: lo
IP Address: 127.0.0.1
Subnet Mask: 8
MAC Address: 00:00:00:00:00:00
----------------------------------------
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ nmap -sP 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 04:58 EDT
Nmap done: 256 IP addresses (0 hosts up) scanned in 103.26 seconds

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 05:00 EDT
Stats: 0:01:19 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Ping Scan Timing: About 77.15% done; ETC: 05:02 (0:00:23 remaining)
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ nmap -sV 192.168.1.0/24 > report.txt
```

```
Open
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 05:04 EDT
Nmap done: 256 IP addresses (0 hosts up) scanned in 104.64 seconds
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ sudo tcpdump -i eth0 -w capture.pcap
[sudo] password for kali:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 b
ytes
^D^C13 packets captured
13 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.

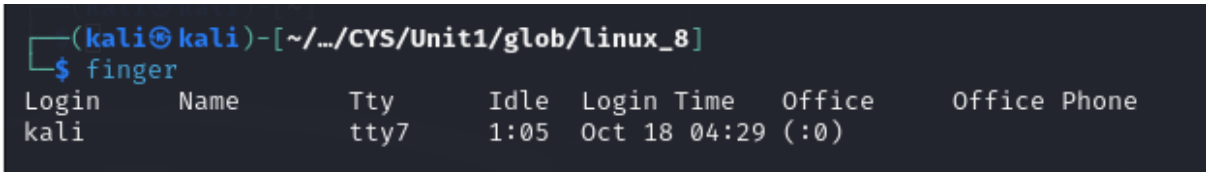
```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ tcpdump -r capture.pcap -A -s 0 'tcp port 80'
reading from file capture.pcap, link-type EN10MB (Ethernet), snapshot length 2621
44
```

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

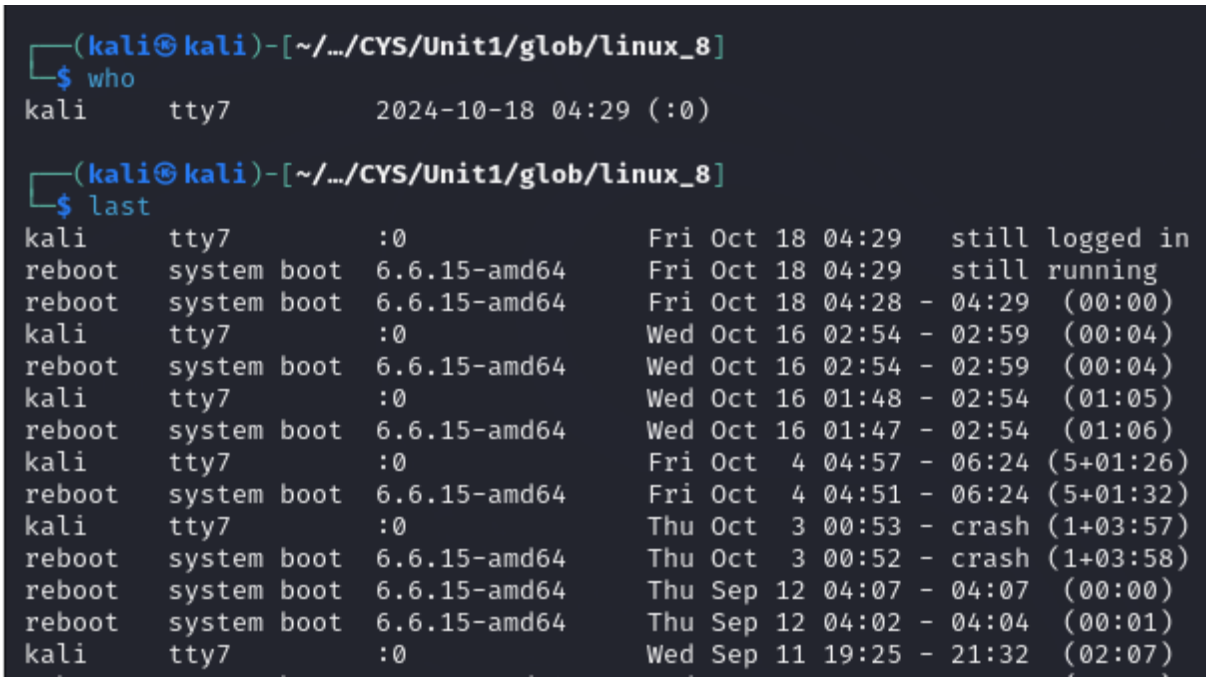
14. Use the hostname command to display and change the hostname of your machine.

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ hostname
kali

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ sudo kali pratik
sudo: kali: command not found

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ sudo hostname pratik

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ hostname
pratik

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ sudo hostname kali
^[[A^[[Asudo: unable to resolve host pratik: Te
^[[A^[[A

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ hostname
kali

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$
```

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



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



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.

```
#!/bin/bash

# Check if a filename is provided
if [ $# -ne 1 ]; then
  echo "Usage: $0 <url_file>"
  exit 1
fi

# Check if the file exists
if [ ! -f "$1" ]; then
  echo "File not found: $1"
  exit 1
fi

# Read each URL from the file and check its status
while IFS= read -r url; do
  if curl -s --head "$url" | grep "200 OK" > /dev/null; then
    echo "$url is up."
  else
    echo "$url is down."
  fi
done < "$1"
```

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ chmod +x testurl.sh
```

```
http://example.com
http://nonexistentwebsite.xyz
https://www.google.com
https://www.github.com
http://thisurldoesnotexist12345.com
```

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ ./testurl.sh urls.txt
http://example.com is up.
http://nonexistentwebsite.xyz is down.
https://www.google.com is down.
https://www.github.com is down.
http://thisurldoesnotexist12345.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.

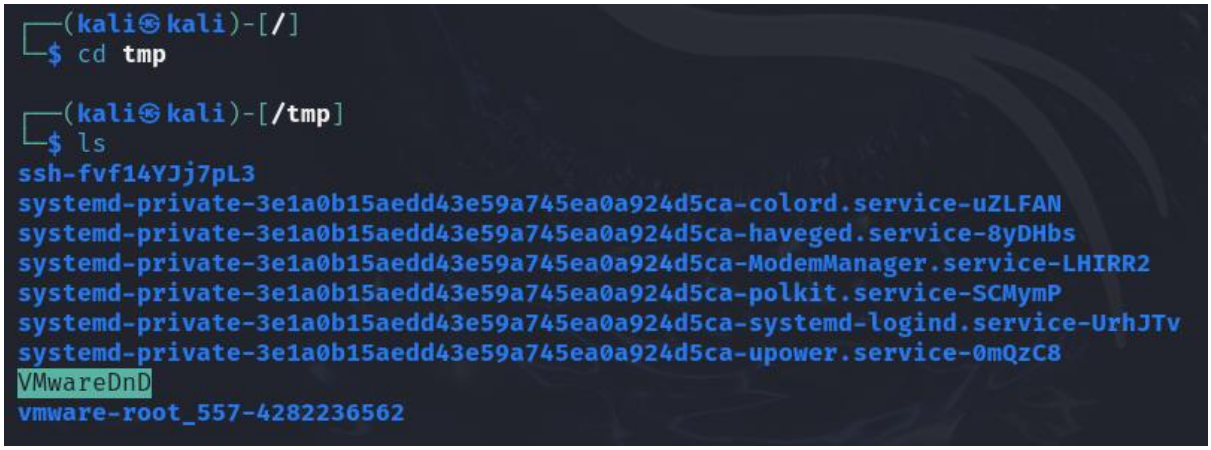
```
#!/bin/bash

# Check if the current directory is specified
if [ "$1" ]; then
  cd "$1" || { echo "Directory not found: $1"; exit 1; }
fi

# List the 5 largest items in the current directory
du -ah . | sort -rh | head -n 5
```

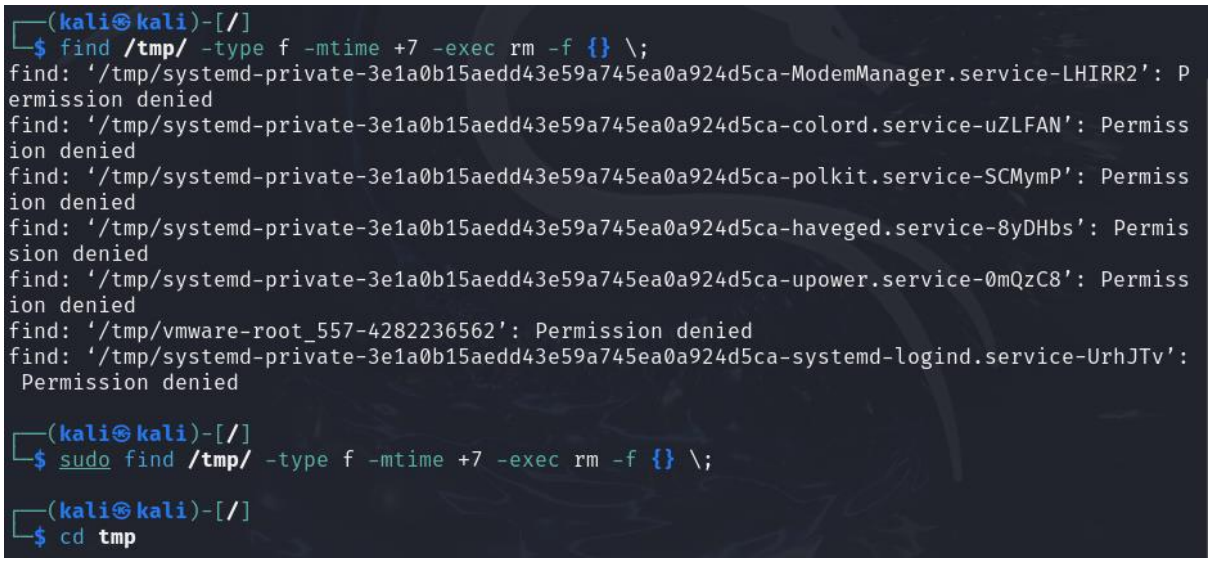
```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ chmod +x diskhog.sh

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/linux_8]
└─$ ./diskhog.sh
32K     .
4.0K    ./urls.txt
4.0K    ./testurl.sh
4.0K    ./report.txt
4.0K    ./lab_8_8.sh
```

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

```
┌──(kali㉿kali)-[/var/log]
└─$ gzip *.log
gzip: alternatives.log.gz: Permission denied
gzip: boot.log: Permission denied
gzip: dpkg.log.gz: Permission denied
gzip: fontconfig.log.gz: Permission denied
gzip: vmware-network.1.log.gz: Permission denied
gzip: vmware-network.2.log.gz: Permission denied
gzip: vmware-network.3.log.gz: Permission denied
gzip: vmware-network.4.log.gz: Permission denied
gzip: vmware-network.5.log.gz: Permission denied
gzip: vmware-network.6.log.gz: Permission denied
gzip: vmware-network.7.log.gz: Permission denied
gzip: vmware-network.8.log.gz: Permission denied
gzip: vmware-network.9.log.gz: Permission denied
gzip: vmware-network.log.gz: Permission denied
gzip: vmware-vmsvc-root.1.log: Permission denied
gzip: vmware-vmsvc-root.2.log: Permission denied
gzip: vmware-vmsvc-root.3.log: Permission denied
gzip: vmware-vmsvc-root.log: Permission denied
gzip: vmware-vmtoolsd-kali.log: Permission denied
gzip: vmware-vmtoolsd-root.log: Permission denied
gzip: vmware-vmusr-kali.log: Permission denied
gzip: Xorg.0.log.gz: Permission denied
gzip: Xorg.1.log.gz: Permission denied
```

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





find /tmp/: Searches within the /tmp/ directory.

-type f: Specifies that you're looking for files (not directories).

-mtime +7: Finds files that were last modified more than 7 days ago.

-exec rm -f {} \;: For each file found, the rm -f command is executed to forcefully delete the file.

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

```
└─$ grep -r "auth" /etc/*.conf
/etc/legion.conf:services="asterisk,afp,cisco,cisco-enable,cvs,firebird,ftp,ftps,http-head,h
ttp-get,https-head,https-get,http-get-form,https-get-form,http-post-form,https-post-form,htt
p-proxy,http-proxy-urlenum,icq,imap,imaps,irc,ldap2,ldap2s,ldap3,ldap3s,ldap3-crammd5,ldap3-
crammd5s,ldap3-digestmd5,ldap3-digestmd5s,mssql,mysql,ncp,nntp,oracle-listener,oracle-sid,pc
anywhere,pcnfs,pop3,pop3s,postgres,rdp,rexec,rlogin,rsh,s7-300,sip,smb,smtp,smtps,smtp-enum,
snmp,socks5,ssh,sshkey,svn,teamspeak,telnet,telnets,vmauthd,vnc,xmpp"
/etc/legion.conf:http-auth-finder.nse=http-auth-finder.nse, "nmap -Pn [IP] -p [PORT] --scrip
t=http-auth-finder.nse --script-args=unsafe=1", "http,https,ssl,soap,http-proxy,http-alt,htt
ps-alt"
/etc/legion.conf:http-auth.nse=http-auth.nse, "nmap -Pn [IP] -p [PORT] --script=http-auth.ns
e --script-args=unsafe=1", "http,https,ssl,soap,http-proxy,http-alt,https-alt"
/etc/legion.conf:realvnc-auth-bypass.nse=realvnc-auth-bypass.nse, "nmap -Pn [IP] -p [PORT] -
-script=realvnc-auth-bypass.nse --script-args=unsafe=1", vnc
/etc/nfs.conf:# debug="all|auth|call|general|parse"
/etc/nfs.conf:# debug="all|auth|call|general|parse"
/etc/proxychains4.conf:#        ( auth types supported: "basic"-http  "user/pass"-socks )
/etc/sudo_logsrvd.conf:# Path to a certificate authority bundle file in PEM format to use
/etc/sudo_logsrvd.conf:# instead of the system's default certificate authority database.
/etc/sudo_logsrvd.conf:# Path to a certificate authority bundle file in PEM format to use
/etc/sudo_logsrvd.conf:# instead of the system's default certificate authority database.
/etc/sudo_logsrvd.conf:# The following syslog facilities are supported: authpriv (if your OS
/etc/sudo_logsrvd.conf:# supports it), auth, daemon, user, local0, local1, local2, local3,
/etc/sudo_logsrvd.conf:#facility = authpriv
/etc/tightvncserver.conf:##  $authType     -- argument to Xvnc specifying authentication type
/etc/tightvncserver.conf:# $authType = "-rfbauth $vncUserDir/passwd";
/etc/tightvncserver.conf:## uncomment the line below. Note that in this case Xvnc's .Xauthor
ity
```

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

```
└─$ grep -i "failed" /var/log/*.log | wc -l
grep: /var/log/boot.log: Permission denied
grep: /var/log/vmware-vmsvc-root.1.log: Permission denied
grep: /var/log/vmware-vmsvc-root.2.log: Permission denied
grep: /var/log/vmware-vmsvc-root.3.log: Permission denied
grep: /var/log/vmware-vmsvc-root.log: Permission denied
grep: /var/log/vmware-vmtoolsd-kali.log: Permission denied
grep: /var/log/vmware-vmtoolsd-root.log: Permission denied
grep: /var/log/vmware-vmusr-kali.log: Permission denied
2
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/bak]
└─$ touch a.txt b.txt c.txt d.txt e.txt

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/bak]
└─$ ls
a.txt  b.txt  c.txt  d.txt  e.txt
```

```
#!/bin/bash

# Loop through all .txt files in the current directory
for file in *.txt; do
    # Check if the file exists (in case there are no .txt files)
    if [[ -e "$file" ]]; then
        mv "$file" "${file%.txt}.bak"
        echo "Renamed '$file' to '${file%.txt}.bak'"
    fi
done
```

S

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/bak]
└─$ ./txt_to_bak.sh
Renamed 'a.txt' to 'a.bak'
Renamed 'b.txt' to 'b.bak'
Renamed 'c.txt' to 'c.bak'
Renamed 'd.txt' to 'd.bak'
Renamed 'e.txt' to 'e.bak'

┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/bak]
└─$ ls
a.bak  b.bak  c.bak  d.bak  e.bak  txt_to_bak.sh
```

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

```
user1
user2
user3
user4
pratik
kali
```

S

Open

```
#!/bin/bash

# Loop through each user in users.txt
while read -r user; do
    if id "$user" &>/dev/null; then
        echo "$user exists."
    else
        echo "$user does not exist."
    fi
done < users.txt
```

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/bak]
└─$ ./check_users.sh
user1 does not exist.
user2 does not exist.
user3 does not exist.
user4 does not exist.
pratik does not exist.
kali exists.
```

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

```
┌──(kali㉿kali)-[~/…/CYS/Unit1/glob/bak]
└─$ find /var/log/ -type f -name "*.log" -size +1M -exec grep -E "ERROR|CRITICAL" {} \;

find: ‘/var/log/lightdm’: Permission denied
find: ‘/var/log/private’: Permission denied
find: ‘/var/log/speech-dispatcher’: Permission denied
find: ‘/var/log/inetsim’: Permission denied
```