CySA+ Lab Series

Lab 04: Linux Command Line Tools

Document Version: 2022-10-10

Material in this Lab Aligns to the Following	
CompTIA CySA+ (CS0-002) Exam Objectives	1.4 - Given a scenario, analyze the output from common vulnerability tools 2.1 - Explain software assurance best practices
All-In-One CompTIA CySA+ Second Edition ISBN-13: 978-1260464306 Chapters	4: Vulnerability Assessment Tools 9: Software Assurance Best Practices

Contents

Introduction

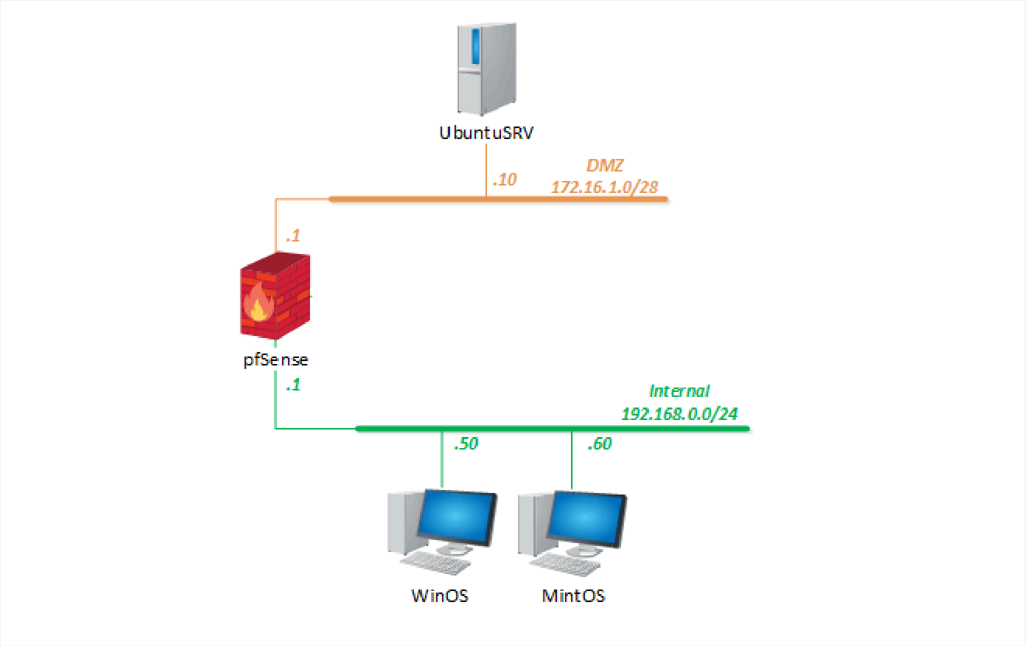
There are many tools that are included with the Linux operating systems that will assist in cybersecurity analysis.

Objective

In this lab, you will explore various Linux command line tools to understand the services and processes running on the Windows system.

- Using *IFCONFIG* and *IP*
- Using *PING*
- Using *TRACEROUTE*
- Identifying Routes
- Identifying Users using *WHOAMI*
- Identifying Nearby Systems Using *ARP*

Lab Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account	Password
WinOS (Server 2019)	`192.168.0.50`	`Administrator`	`NDGlabpass123!`
MintOS (Linux Mint)	`192.168.0.60`	`sysadmin`	`NDGlabpass123!`
OSSIM (Alien Vault)	`172.16.1.2`	`root`	`NDGlabpass123!`
UbuntuSRV (Ubuntu Server)	`172.16.1.10`	`sysadmin`	`NDGlabpass123!`
Kali	`203.0.113.2`	`sysadmin`	`NDGlabpass123!`

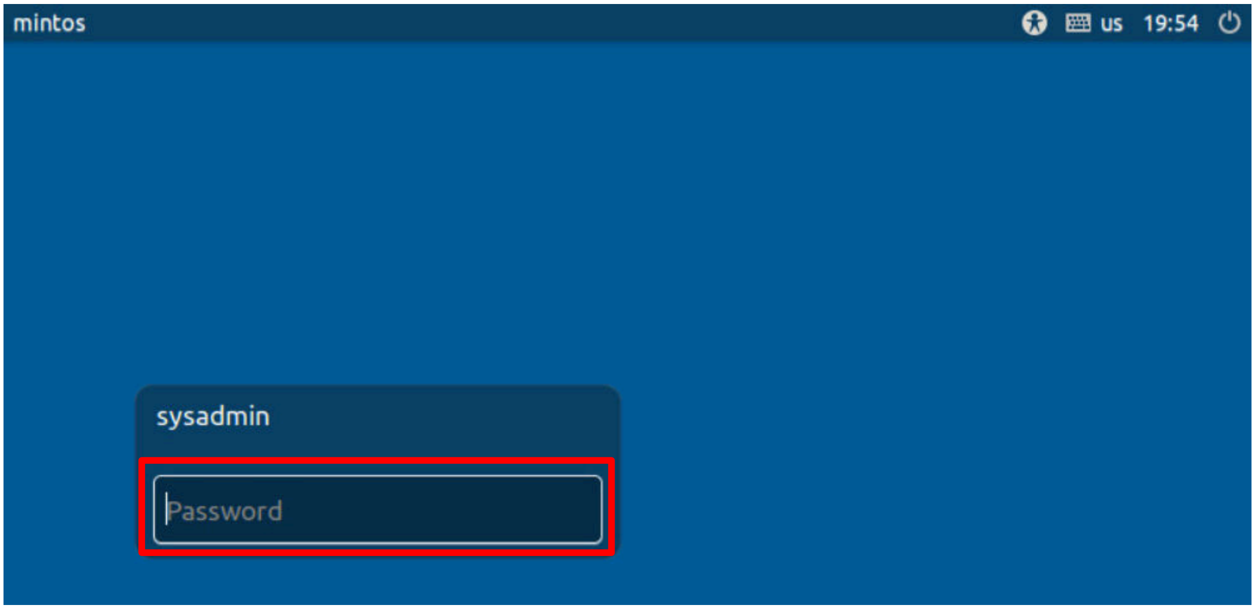
pfSense	`203.0.113.1`
	`172.16.1.1`
	`192.168.0.1`
		`admin`	`NDGlabpass123!`

1 Using IFCONFIG and IP

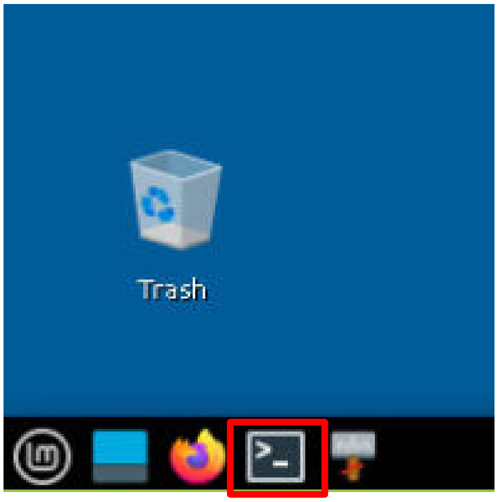
1.1 ifconfig Command

ifconfig is a console application that displays information about the IPv4 and IPv6 stack on a Linux or Unix computer.

1. Click on the **MintOS** tab to access the graphical login screen.
2. Log in as *sysadmin* using the password: `NDGlabpass123!`

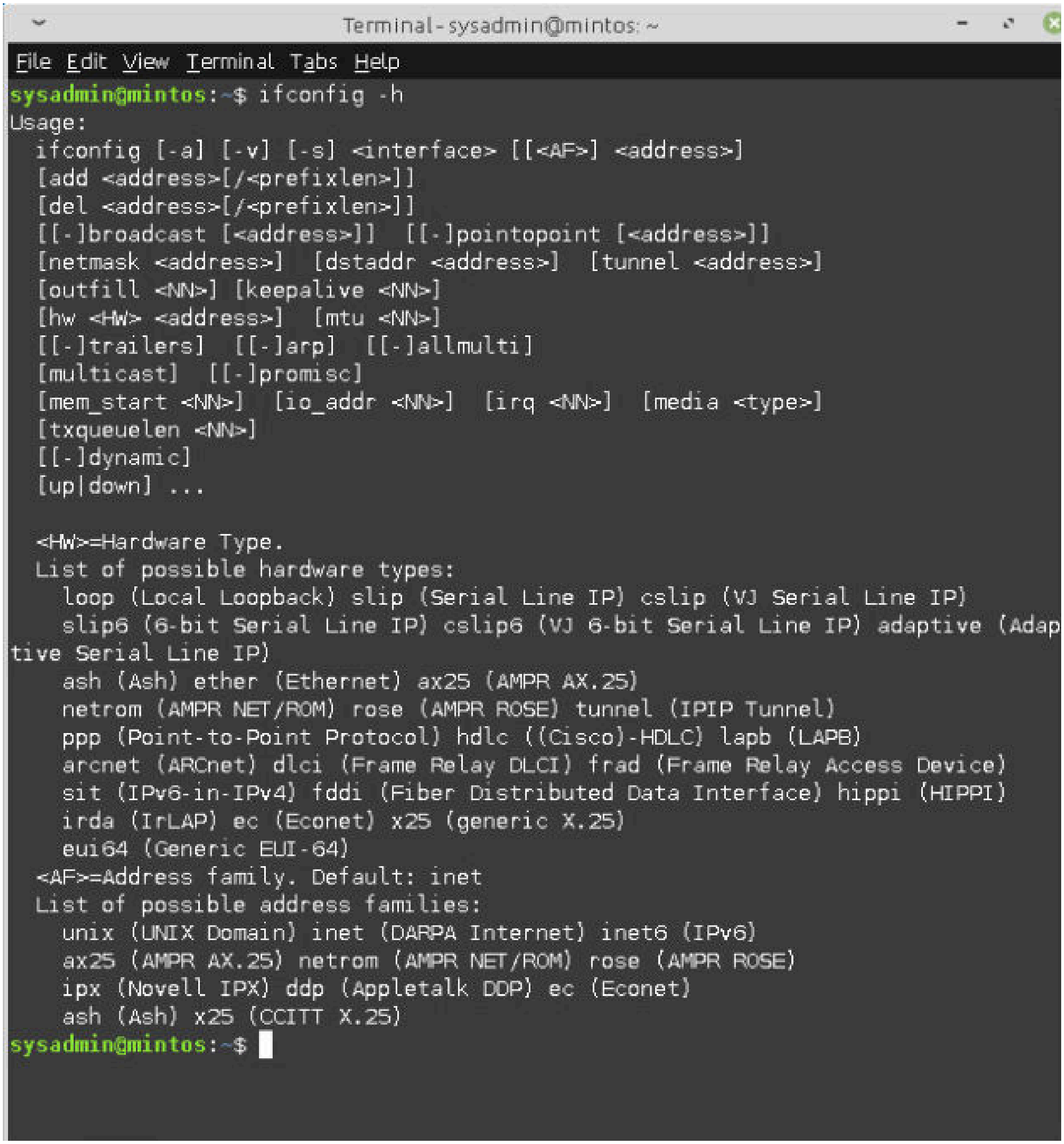


3. Click on the **Terminal** icon in the taskbar at the bottom of the screen.



4. In the command prompt window, type the following command to show the *ifconfig* help screen:

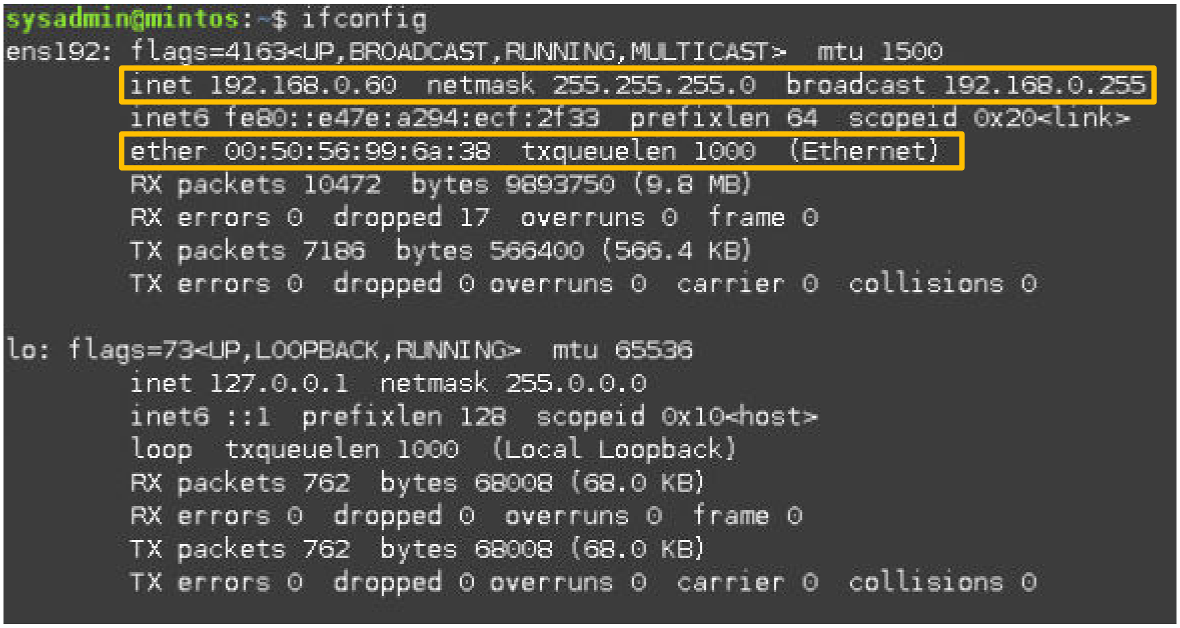
```
ifconfig -h
```


Terminal - sysadmin@mintos: ~
File Edit View Terminal Tabs Help

```
sysadmin@mintos:~$ ifconfig -h
Usage:
  ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
  [add <address>[/<prefixlen>]]
  [del <address>[/<prefixlen>]]
  [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
  [netmask <address>]  [dstaddr <address>]  [tunnel <address>]
  [outfill <NN>] [keepalive <NN>]
  [hw <HW> <address>]  [mtu <NN>]
  [[-]trailers]  [[-]arp]  [[-]allmulti]
  [multicast]  [[-]promisc]
  [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]  [media <type>]
  [txqueuelen <NN>]
  [[-]dynamic]
  [up|down] ...

  <HW>=Hardware Type.
  List of possible hardware types:
    loop (Local Loopback) slip (Serial Line IP) cslip (VJ Serial Line IP)
    slip6 (6-bit Serial Line IP) cslip6 (VJ 6-bit Serial Line IP) adaptive (Adap
tive Serial Line IP)
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) tunnel (IPIP Tunnel)
    ppp (Point-to-Point Protocol) hdlc ((Cisco)-HDLC) lapb (LAPB)
    arcnet (ARCnet) dlci (Frame Relay DLCI) frad (Frame Relay Access Device)
    sit (IPv6-in-IPv4) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) ec (Econet) x25 (generic X.25)
    eui64 (Generic EUI-64)
  <AF>=Address family. Default: inet
  List of possible address families:
    unix (UNIX Domain) inet (DARPA Internet) inet6 (IPv6)
    ax25 (AMPR AX.25) netrom (AMPR NET/ROM) rose (AMPR ROSE)
    ipx (Novell IPX) ddp (Appletalk DDP) ec (Econet)
    ash (Ash) x25 (CCITT X.25)
sysadmin@mintos:~$
```

5. To show the *IP address, subnet mask, default gateway,* and more information for all the active adapters, type the following command:

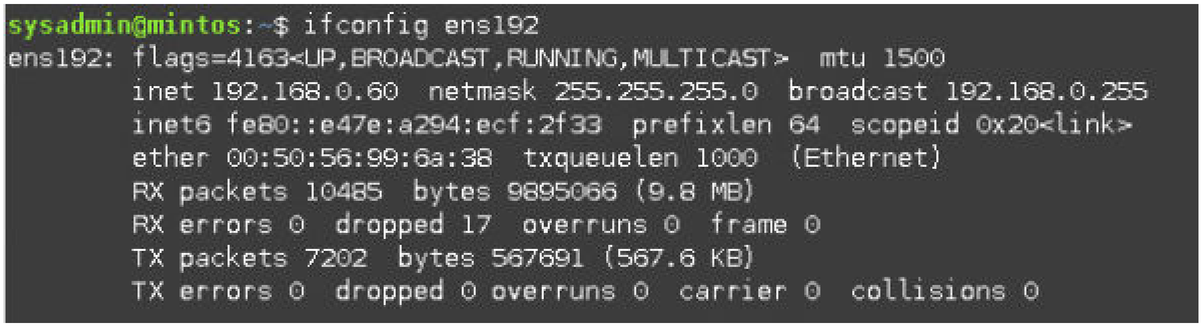
```
ifconfig
```



```
sysadmin@mintos:~$ ifconfig
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.60  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::e47e:a294:ecf:2f33  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:99:6a:38  txqueuelen 1000  (Ethernet)
        RX packets 10472  bytes 9893750 (9.8 MB)
        RX errors 0  dropped 17  overruns 0  frame 0
        TX packets 7186  bytes 566400 (566.4 KB)
        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 762  bytes 68008 (68.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 762  bytes 68008 (68.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

6. To show the settings for a specific interface, type the following command:

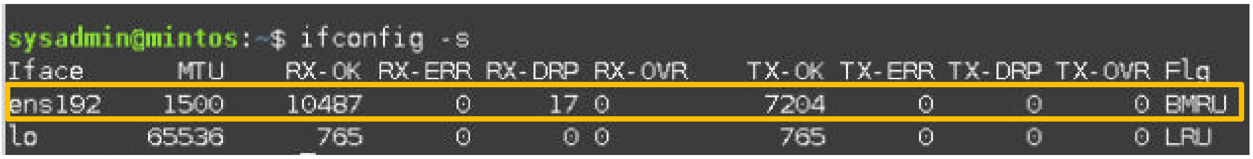
```
ifconfig ens192
```



```
sysadmin@mintos:~$ ifconfig ens192
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.60  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::e47e:a294:ecf:2f33  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:99:6a:38  txqueuelen 1000  (Ethernet)
        RX packets 10485  bytes 9895066 (9.8 MB)
        RX errors 0  dropped 17  overruns 0  frame 0
        TX packets 7202  bytes 567691 (567.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

7. To show the list of network adapters along with their respective media status, type the following command:

```
ifconfig -s
```



```
sysadmin@mintos:~$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens192    1500    10487      0     17 0          7204      0      0      0 BMRU
lo       65536      765      0      0 0           765      0      0      0 LRU
```

8. Leave the *MintOS* machine open and continue to the next task.

1.2 ip Command

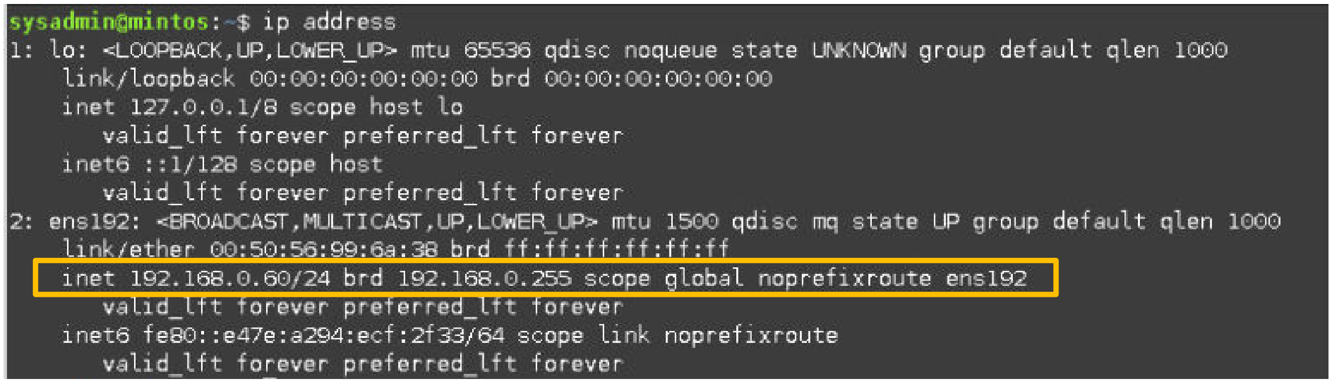
The *ifconfig* command has been deprecated and has been replaced by the more powerful *ip* command, which is much broader in functionality.

The *ip* command can perform the following additional tasks:

- Displaying or Modifying Interface properties.
- Adding, Removing ARP Cache entries along with creating new Static ARP entries for a host.
- Displaying MAC addresses associated with all the interfaces.
- Displaying and modifying kernel routing tables.

1. The *ifconfig* command displays only the enabled interfaces, whereas the *ip* command will show all of the interfaces, enabled or not. To display all of the interfaces and their status, type the following command:

```
ip address
```



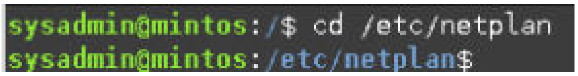
```
sysadmin@mintos:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:99:6a:38 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.60/24 brd 192.168.0.255 scope global noprefixroute ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::e47e:a294:ecf:2f33/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

You can abbreviate the command using **ip a** instead of **ip address**.

If you look at the highlighted output, the statement **global nonprefixroute ens192**, indicates that this is a static address. If the address was assigned through **DHCP,** it would display **global <u>dynamic</u> nonprefixroute ens192**. Make a note of the **inet address** of the **ens192** interface, which is **192.168.0.60**. You will use it for the next step.

2. The IP address was set up with a static address **192.168.0.60/24**. To change it to a DHCP assigned address, we need to edit the configuration file. Change to the **netplan** directory by typing in the following command:

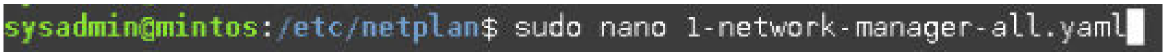
```
cd /etc/netplan
```



```
sysadmin@mintos:/$ cd /etc/netplan
sysadmin@mintos:/etc/netplan$
```

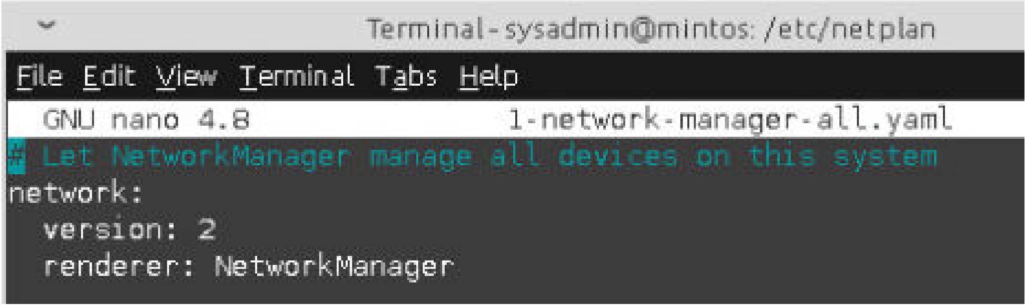
3. Change the configuration file **1-network-manager-all.yaml** using the **nano** editor as root. When asked for the **[sudo] password**, type: `NDGlabpass123!`

```
sudo nano 1-network-manager-all.yaml
```



```
sysadmin@mintos:/etc/netplan$ sudo nano 1-network-manager-all.yaml
```

4. The nano editor for the file should look like the screen below:



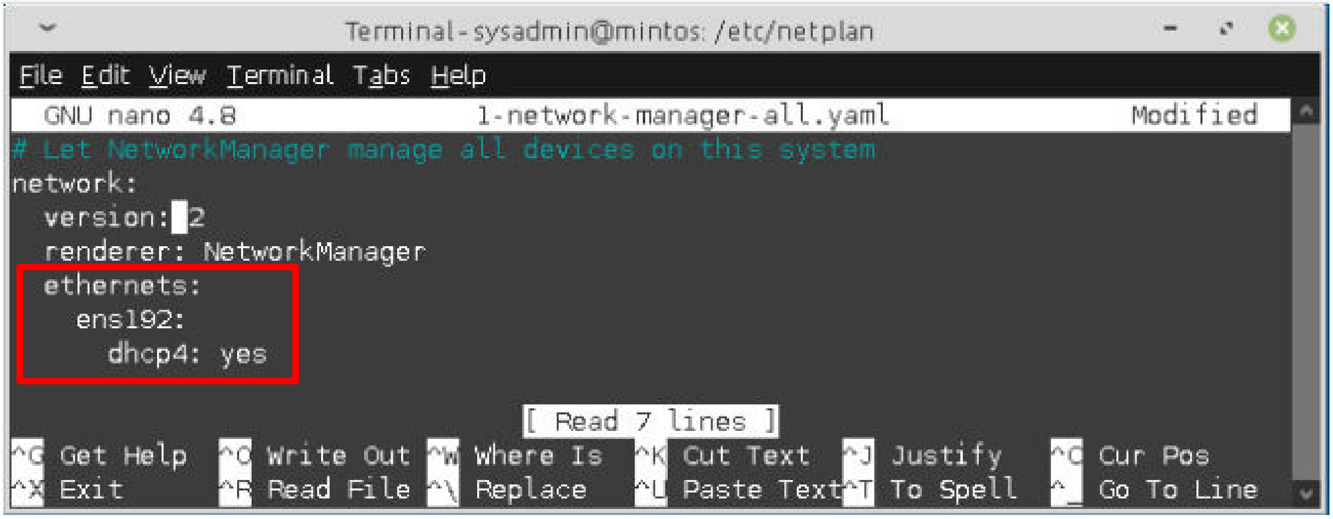
Terminal - sysadmin@mintos: /etc/netplan
File Edit View Terminal Tabs Help

```
  GNU nano 4.8                 1-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
```

Add the following additional lines to the bottom of the file:

```
  ethernets:
    ens192:
      dhcp4: yes
```

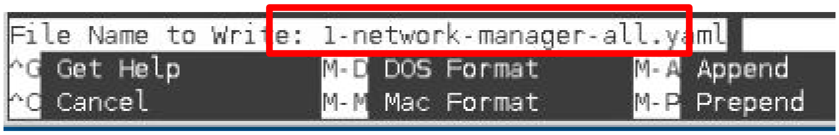
The complete file should look like the screen below:



Terminal - sysadmin@mintos: /etc/netplan
File Edit View Terminal Tabs Help

```
  GNU nano 4.8                1-network-manager-all.yaml                    Modified
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens192:
      dhcp4: yes

                              [ Read 7 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

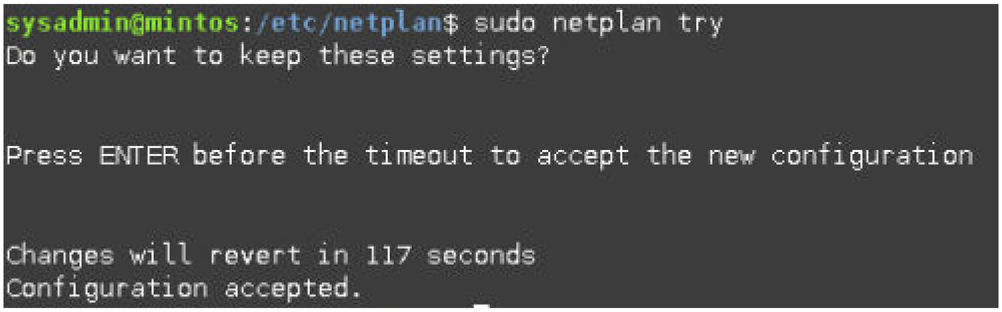
5. Type **CTRL-O** to write out the file, press **Enter** to use the same file name, and then type **CTRL-X** to exit.



```
File Name to Write: 1-network-manager-all.yaml
^G Get Help          M-D DOS Format       M-A Append
^C Cancel            M-M Mac Format       M-P Prepend
```

6. Test the configuration file by typing the following command. If the configuration is accepted, press **Enter** to accept the settings. If the test fails, the configuration will be rejected.

```
sudo netplan try
```



```
sysadmin@mintos:/etc/netplan$ sudo netplan try
Do you want to keep these settings?


Press ENTER before the timeout to accept the new configuration


Changes will revert in 117 seconds
Configuration accepted.
```

Press **Enter** to accept the configuration.

7. Apply the configuration by typing the following command:

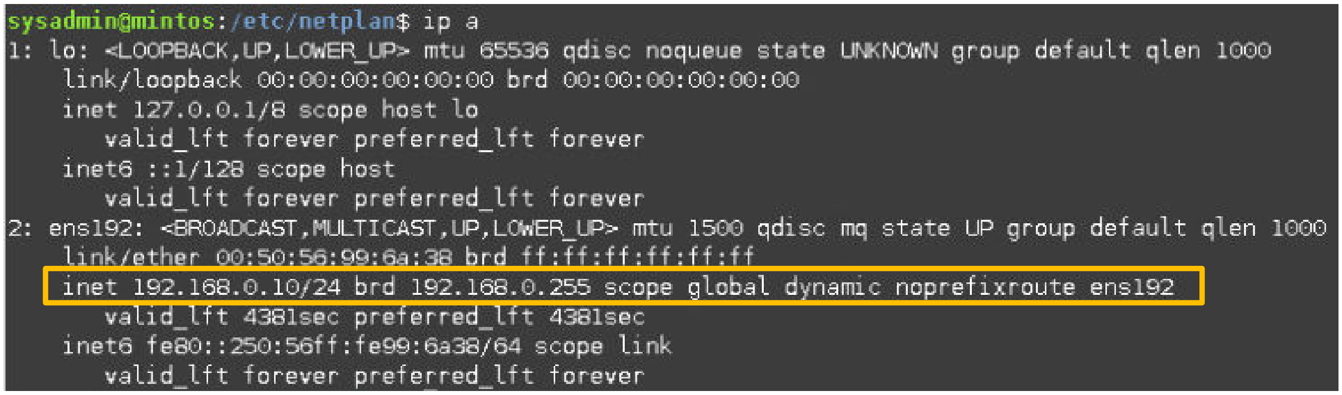
```
sudo netplan apply
```

8. If the configuration has been successfully applied, restart the **Network Manager** service by typing the following command:

```
sudo systemctl restart network-manager
```

9. Verify that the address has been changed by typing the **ip a** command:

```
ip a
```



```
sysadmin@mintos:/etc/netplan$ 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
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:99:6a:38 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.10/24 brd 192.168.0.255 scope global dynamic noprefixroute ens192
       valid_lft 4381sec preferred_lft 4381sec
    inet6 fe80::250:56ff:fe99:6a38/64 scope link
       valid_lft forever preferred_lft forever
```

The actual IP address might be different than the example, due to DHCP. However, the address should be on Network Address 192.168.0.0/24 and the host should not be 60.

10. Leave the *MintOS* machine open and continue to the next task.

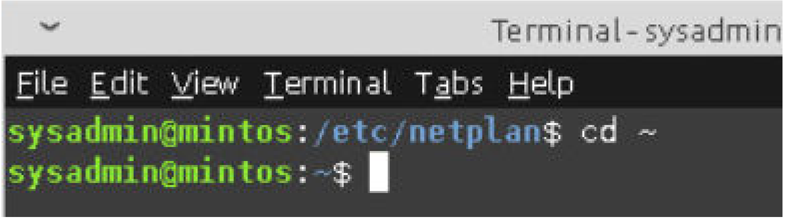
2 Using PING

The *ping* command is used to verify IP level connectivity between hosts by sending out an **ICMP (Internet Control Message Protocol) Echo Request** packet to the other host's IP address which then returns an **ICMP Echo Reply** packet. The results of the reply are displayed on the console screen. Part of the reply contains information about the amount of time it takes for the round trip, which can inform you about your network's latency time.

The *ping* command is a TCP/IP command and is available on many different operating systems, including Windows, macOS, and Linux/Unix. It can troubleshoot connectivity, reachability and name resolution issues. You can ping a host by IP address or by hostname. If you can ping a host by IP address, but not by name, you know you have a problem with name resolution using DNS (Domain Name System) or the local HOSTS file.

1. In the terminal, return to the home directory by typing:

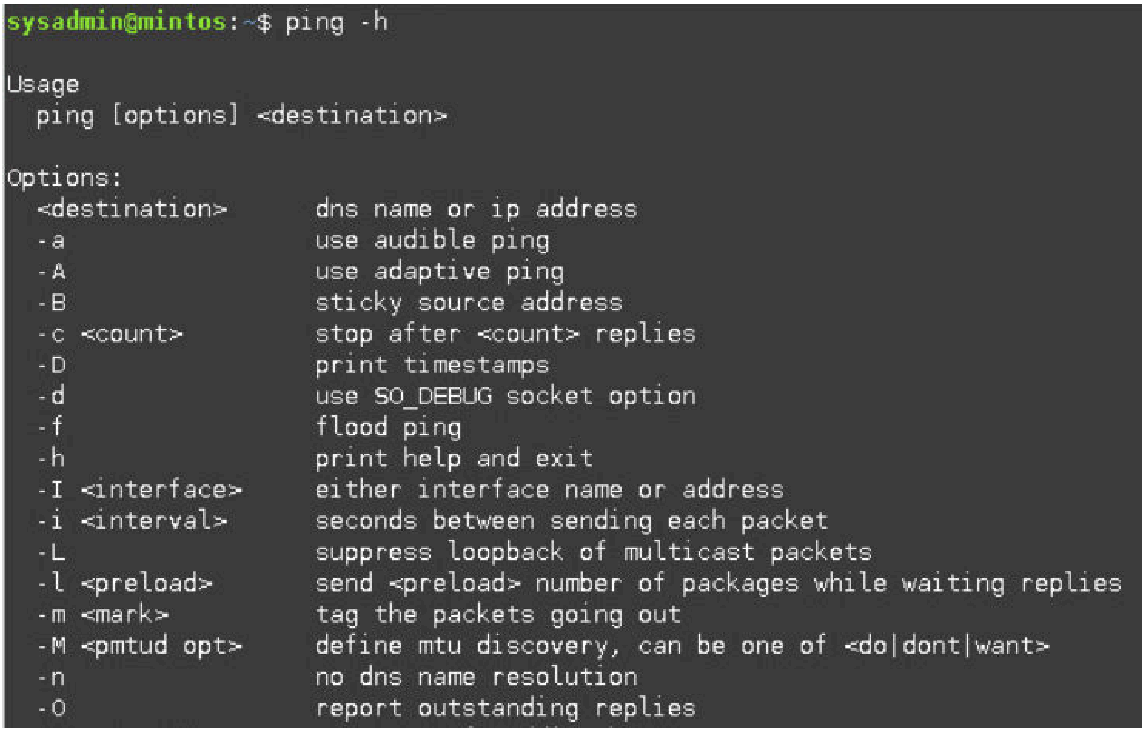
```
cd ~
```



```
sysadmin@mintos:/etc/netplan$ cd ~
sysadmin@mintos:~$
```

2. In the terminal window, type the following command to show the *ping* help screen.

```
ping -h
```

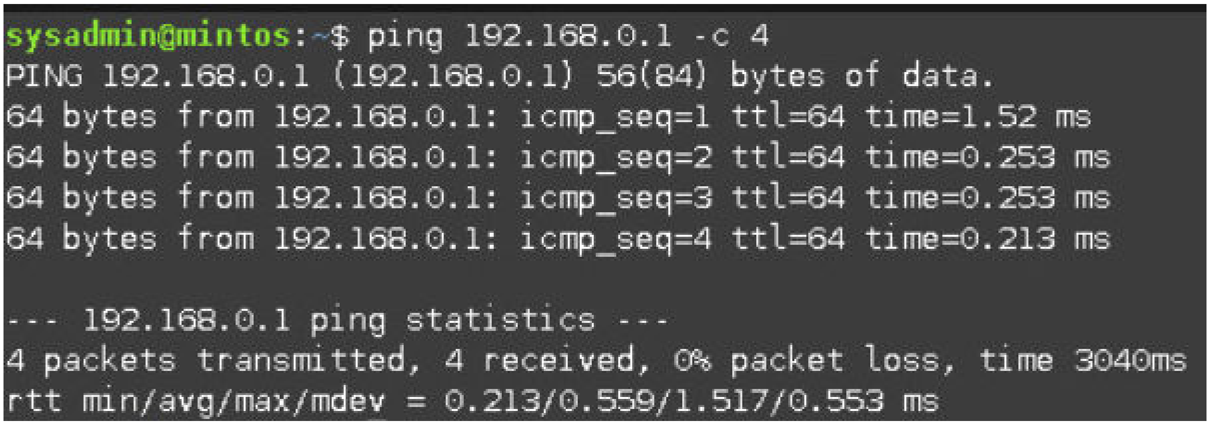


```
sysadmin@mintos:~$ ping -h

Usage
  ping [options] <destination>

Options:
  <destination>      dns name or ip address
  -a                 use audible ping
  -A                 use adaptive ping
  -B                 sticky source address
  -c <count>         stop after <count> replies
  -D                 print timestamps
  -d                 use SO_DEBUG socket option
  -f                 flood ping
  -h                 print help and exit
  -I <interface>     either interface name or address
  -i <interval>      seconds between sending each packet
  -L                 suppress loopback of multicast packets
  -l <preload>       send <preload> number of packages while waiting replies
  -m <mark>          tag the packets going out
  -M <pmtud opt>     define mtu discovery, can be one of <do|dont|want>
  -n                 no dns name resolution
  -O                 report outstanding replies
```

3. To execute the basic *ping* command, type the following command:

```
ping 192.168.0.1 -c 4
```

An important value to look at is the **Time** (in milliseconds). This can be used to determine latency which can be useful in troubleshooting connectivity.

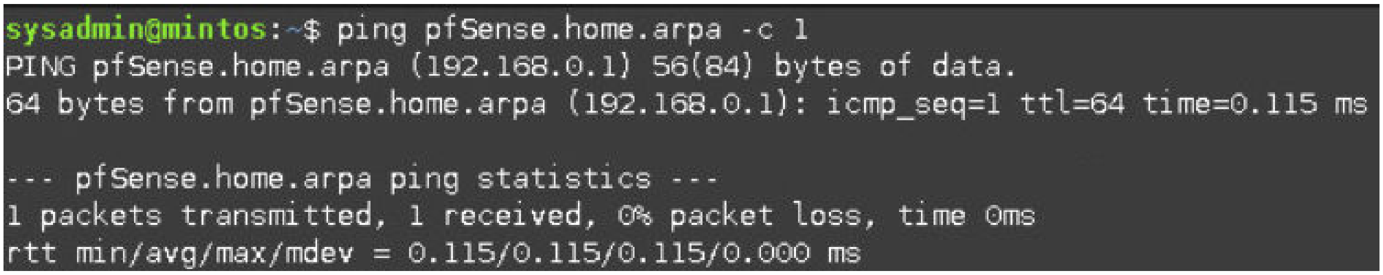


```
sysadmin@mintos:~$ ping 192.168.0.1 -c 4
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=1.52 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.253 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.253 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.213 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3040ms
rtt min/avg/max/mdev = 0.213/0.559/1.517/0.553 ms
```

When you execute the *ping* command in Linux or Unix without using the **-c** option, the pings will continue until you manually stop it with **CTRL+C.**

4. If you want to ping a host by name, type the following command:

```
ping pfSense.home.arpa -c 1
```



```
sysadmin@mintos:~$ ping pfSense.home.arpa -c 1
PING pfSense.home.arpa (192.168.0.1) 56(84) bytes of data.
64 bytes from pfSense.home.arpa (192.168.0.1): icmp_seq=1 ttl=64 time=0.115 ms

--- pfSense.home.arpa ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.115/0.115/0.115/0.000 ms
```

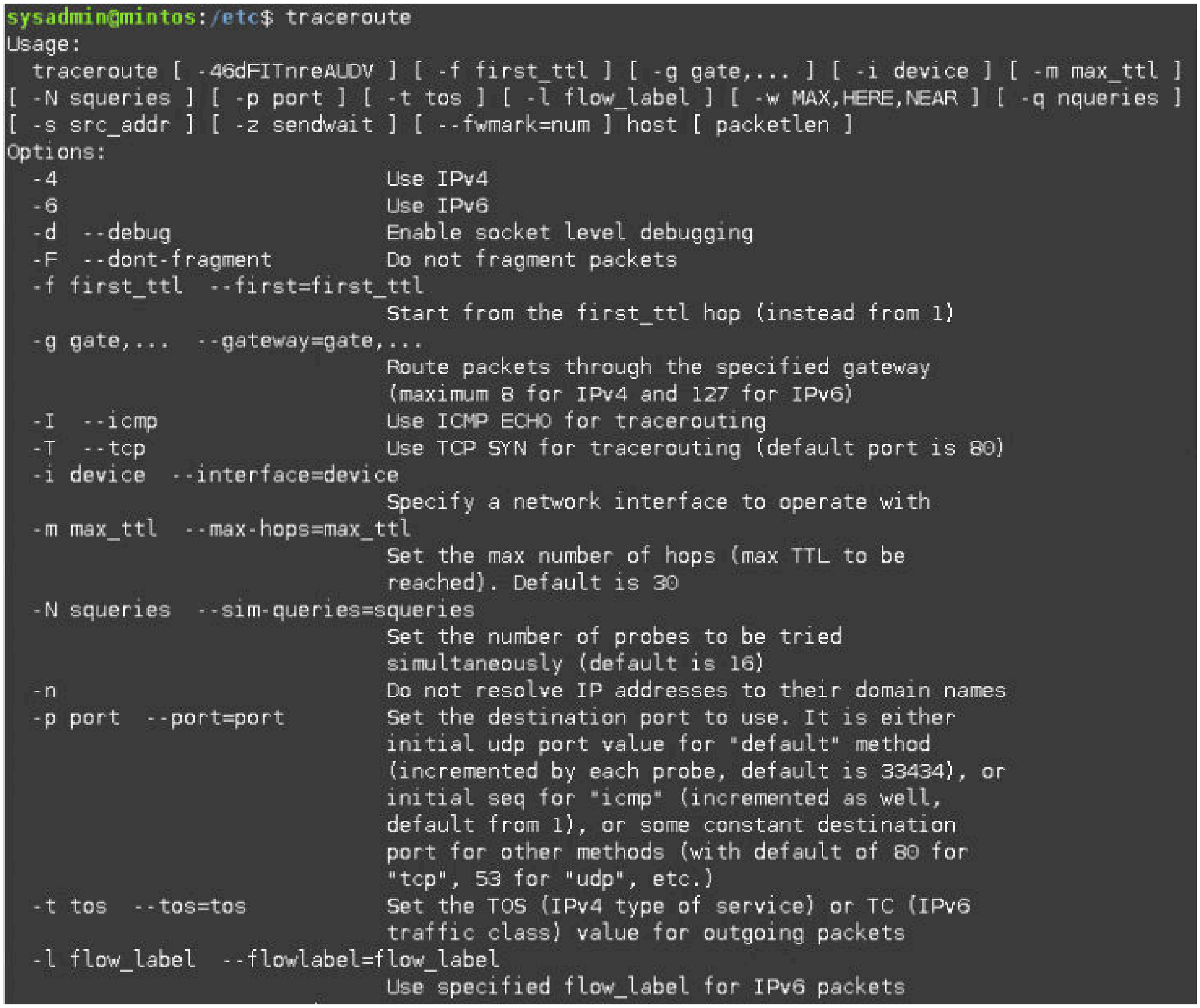
5. Leave the *MintOS* machine open and continue to the next task.

3 Using TRACEROUTE

The *traceroute* command is used to determine the path that is taken to reach a destination host address. In Windows OS, it uses a succession of **ICMP Echo Request** (ping) packets with modified TTL values to determine the IP address and name for each router and network that the ping packet traverse. **macOS** and **Linux** use **UDP** by default, but you can choose a variety of protocols, including **ICMP**, **TCP,** and **GRE**.

1. In the command prompt window, type the following command to show the *traceroute* help screen.

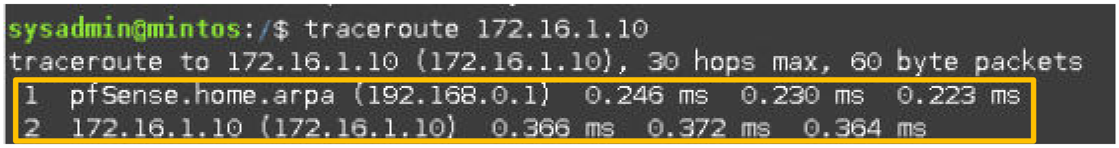
```
traceroute
```



```
sysadmin@mintos:/etc$ traceroute
Usage:
  traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ]
[ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,HERE,NEAR ] [ -q nqueries ]
[ -s src_addr ] [ -z sendwait ] [ --fwmark=num ] host [ packetlen ]
Options:
  -4                          Use IPv4
  -6                          Use IPv6
  -d  --debug                 Enable socket level debugging
  -F  --dont-fragment         Do not fragment packets
  -f first_ttl  --first=first_ttl
                              Start from the first_ttl hop (instead from 1)
  -g gate,...  --gateway=gate,...
                              Route packets through the specified gateway
                              (maximum 8 for IPv4 and 127 for IPv6)
  -I  --icmp                  Use ICMP ECHO for tracerouting
  -T  --tcp                   Use TCP SYN for tracerouting (default port is 80)
  -i device  --interface=device
                              Specify a network interface to operate with
  -m max_ttl  --max-hops=max_ttl
                              Set the max number of hops (max TTL to be
                              reached). Default is 30
  -N squeries  --sim-queries=squeries
                              Set the number of probes to be tried
                              simultaneously (default is 16)
  -n                          Do not resolve IP addresses to their domain names
  -p port  --port=port        Set the destination port to use. It is either
                              initial udp port value for "default" method
                              (incremented by each probe, default is 33434), or
                              initial seq for "icmp" (incremented as well,
                              default from 1), or some constant destination
                              port for other methods (with default of 80 for
                              "tcp", 53 for "udp", etc.)
  -t tos  --tos=tos           Set the TOS (IPv4 type of service) or TC (IPv6
                              traffic class) value for outgoing packets
  -l flow_label  --flowlabel=flow_label
                              Use specified flow_label for IPv6 packets
```

2. Let's test it out by sending the *traceroute* command to the *UbuntuSRV* host at **172.16.1.10** by typing the following command:

```
traceroute 172.16.1.10
```



```
sysadmin@mintos:/$ traceroute 172.16.1.10
traceroute to 172.16.1.10 (172.16.1.10), 30 hops max, 60 byte packets
 1  pfSense.home.arpa (192.168.0.1)  0.246 ms  0.230 ms  0.223 ms
 2  172.16.1.10 (172.16.1.10)  0.366 ms  0.372 ms  0.364 ms
```

The *traceroute* command can be executed from the *MintOS* machine, but since NETLAB+ pods cannot access the internet, the *MintOS* system is blocked from being able to find the hops to any internet address.

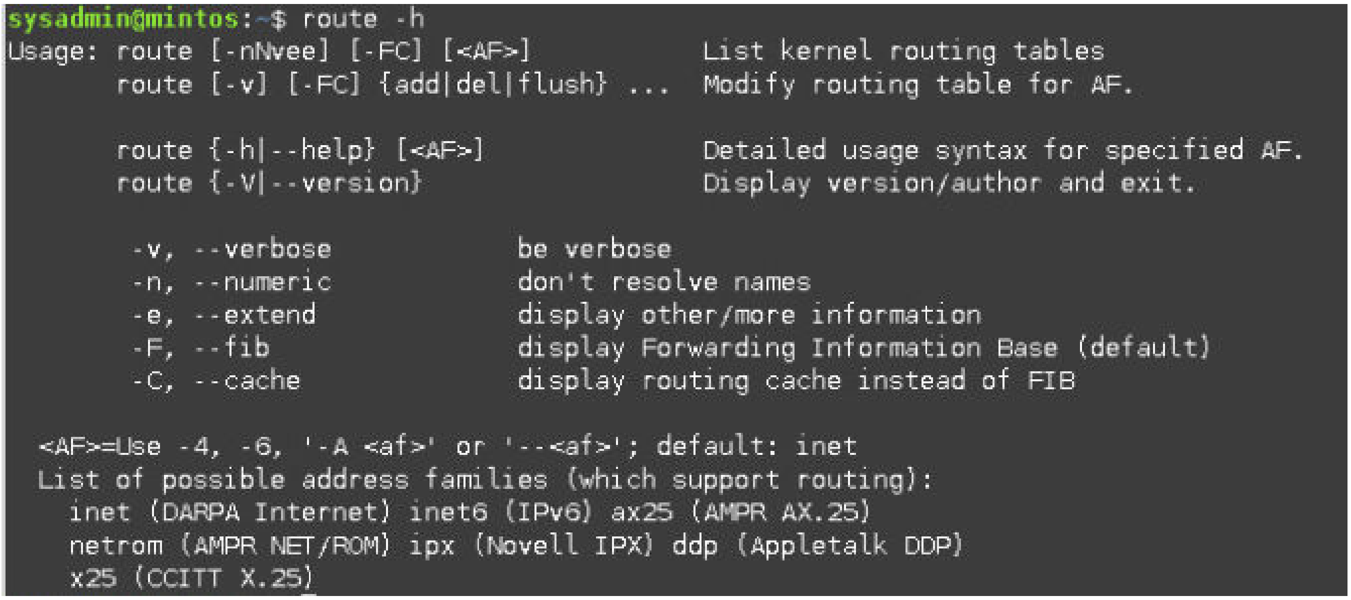
3. Leave the *MintOS* machine open and continue to the next task.

4 Using ROUTE

The *route* command displays and can allow modification to the routing table on the local host.

1. In the command prompt window, type the following command to show the *route* help screen:

```
route -h
```



```
sysadmin@mintos:~$ route -h
Usage: route [-nNvee] [-FC] [<AF>]           List kernel routing tables
       route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.

       route {-h|--help} [<AF>]              Detailed usage syntax for specified AF.
       route {-V|--version}                  Display version/author and exit.

        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -e, --extend             display other/more information
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB

  <AF>=Use -4, -6, '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

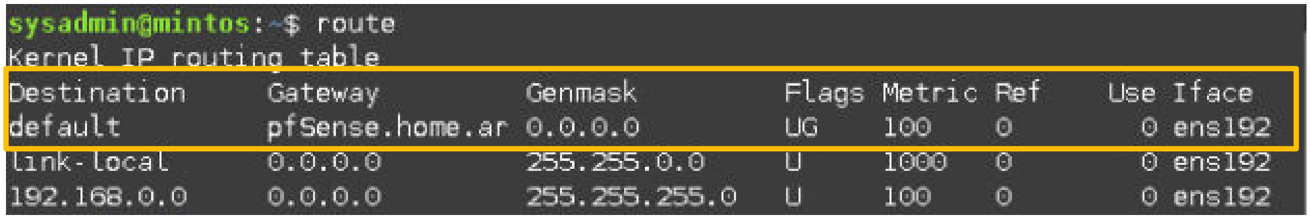
2. To display the routing table, type the following command:

```
route
```

The columns in the routing table are:

Network Destination	The Network IP address or name of a remote network.
Gateway	The IP address or name of the default gateway, or the "next hop" to direct the packet off to the router, which will route the packet to the next network.
Genmask	The network or subnet mask, shown in dotted notation.
Flags	Shows the status of the interface. The flags are: **U**: The route is up **G:** The route is a gateway. No G indicates the route is directly connected **H**: The route is to a host address. No H indicates the route is to a network **D**: The route is created by a redirect **M**: The route is modified by redirect.

Metric	Is the distance, usually counted in hops, to the destination network. Not used in recent Linux kernels
Ref	Number of references to this route. Not used in the Linux Kernel.
Use	Count of lookups for the route.
Iface	The interface to which packets for this route will be sent.

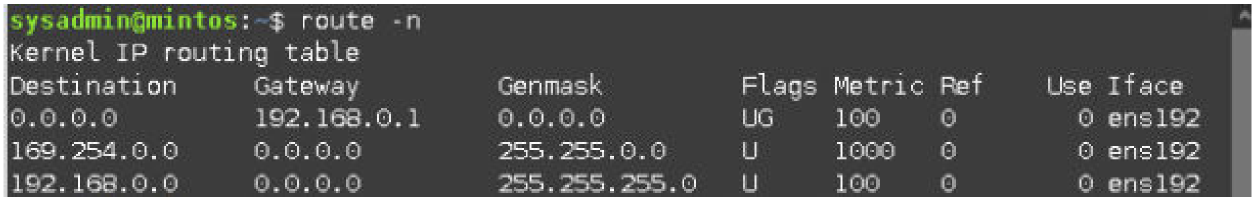


```
sysadmin@mintos:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         pfSense.home.ar 0.0.0.0         UG    100    0        0 ens192
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 ens192
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 ens192
```

The **default** destination shows the **Default Gateway**, which is the next hop if there are no destination matches in the routing table.

3. To display the routing table in full dotted numeric format, type the following command:

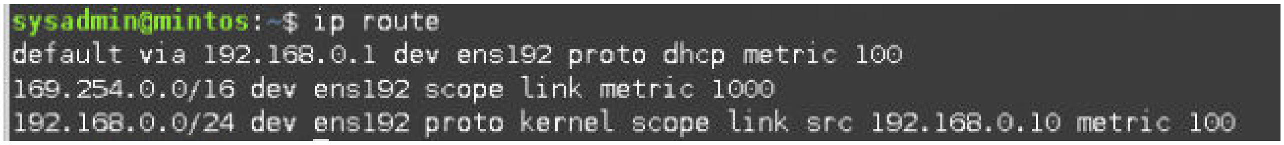
```
route -n
```



```
sysadmin@mintos:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 ens192
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 ens192
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 ens192
```

An alternative method to show the routing table is to use the **ip route** command:

```
ip route
```



```
sysadmin@mintos:~$ ip route
default via 192.168.0.1 dev ens192 proto dhcp metric 100
169.254.0.0/16 dev ens192 scope link metric 1000
192.168.0.0/24 dev ens192 proto kernel scope link src 192.168.0.10 metric 100
```

4. Leave the *MintOS* machine open and continue to the next task.

5 Using ARP and IP NEIGHBOR

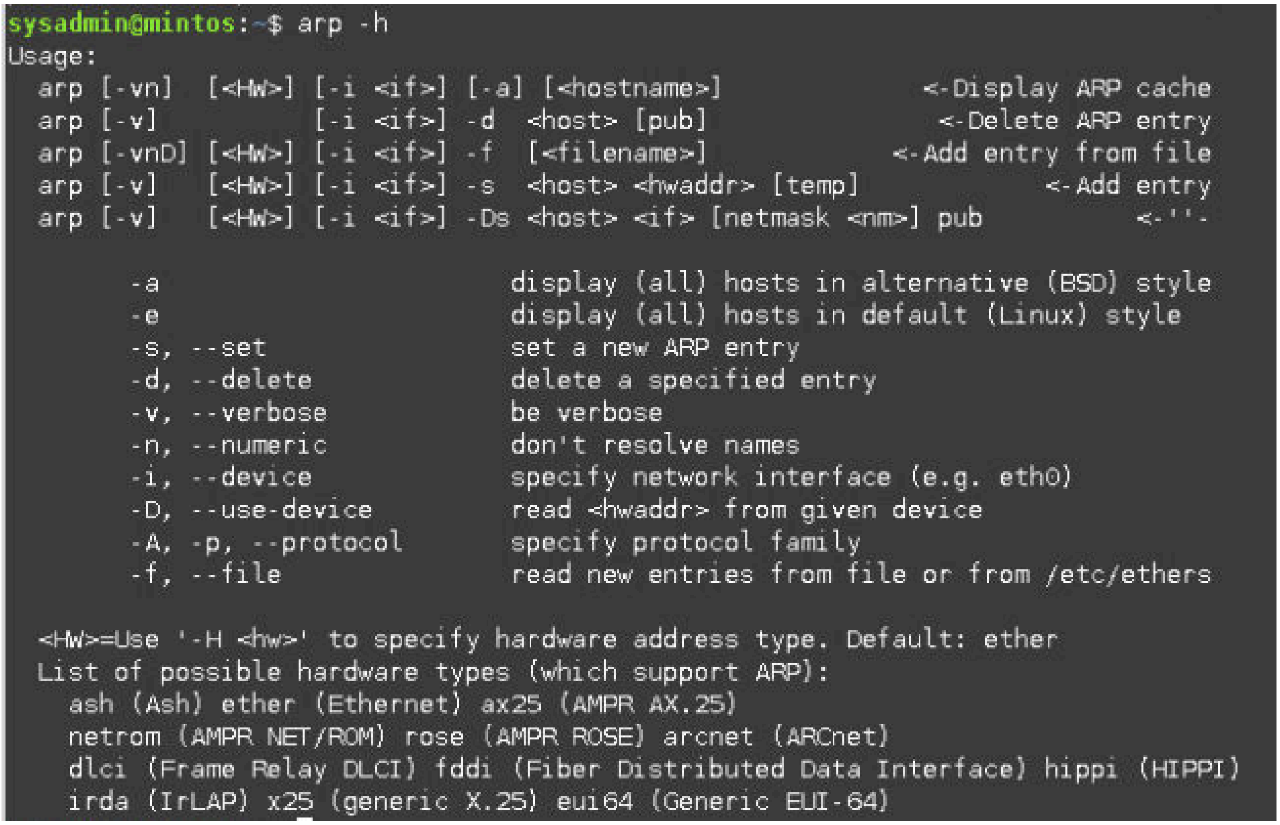
5.1 arp Command

ARP (Address Resolution Protocol) is the protocol that is used to bridge between OSI Layer 2 protocols, such as Ethernet and WIFI, and OSI Layer 3 protocols, primarily IP. It is used to map Layer 2 MAC addresses with their corresponding IP addresses. The tables are kept in the memory of the hosts that reside on a local network. The table is built dynamically as hosts are contacted (such as by a *ping*).

The *ARP* protocol is used for host-to-host connectivity and rarely requires human intervention. But, it is important to monitor the status of the ARP tables because they can be compromised by hackers and become a vector for malware.

1. In the command prompt window, type the following command to show the *arp* help screen.

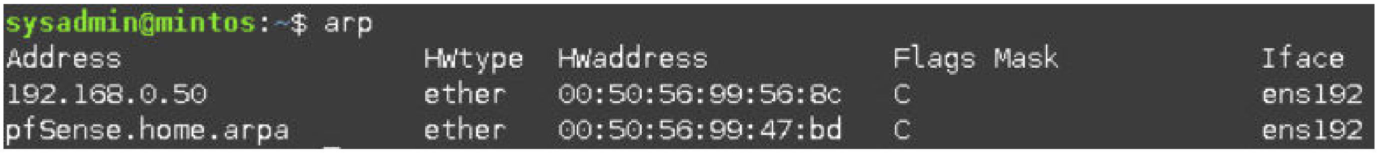
```
arp -h
```



```
sysadmin@mintos:~$ arp -h
Usage:
  arp [-vn]  [<HW>] [-i <if>] [-a] [<hostname>]             <-Display ARP cache
  arp [-v]          [-i <if>] -d  <host> [pub]               <-Delete ARP entry
  arp [-vnD] [<HW>] [-i <if>] -f  [<filename>]            <-Add entry from file
  arp [-v]   [<HW>] [-i <if>] -s  <host> <hwaddr> [temp]            <-Add entry
  arp [-v]   [<HW>] [-i <if>] -Ds <host> <if> [netmask <nm>] pub          <-''-

        -a                       display (all) hosts in alternative (BSD) style
        -e                       display (all) hosts in default (Linux) style
        -s, --set                set a new ARP entry
        -d, --delete             delete a specified entry
        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -i, --device             specify network interface (e.g. eth0)
        -D, --use-device         read <hwaddr> from given device
        -A, -p, --protocol       specify protocol family
        -f, --file               read new entries from file or from /etc/ethers

  <HW>=Use '-H <hw>' to specify hardware address type. Default: ether
  List of possible hardware types (which support ARP):
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
    dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
    irda (IrLAP) x25 (generic X.25) eui64 (Generic EUI-64)
```

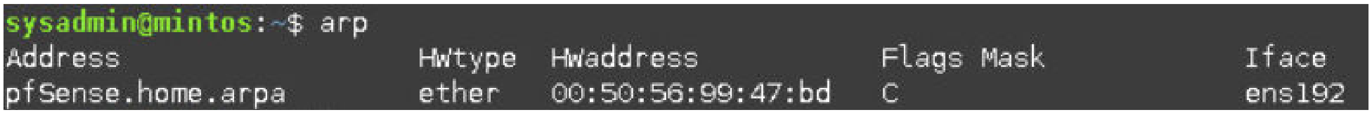
2. To display the *ARP* table for the interfaces on the host, type the following command:

```
arp
```



```
sysadmin@mintos:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.50             ether   00:50:56:99:56:8c   C                     ens192
pfSense.home.arpa        ether   00:50:56:99:47:bd   C                     ens192
```

You might not see the *WinOS* computer in the *ARP* table.

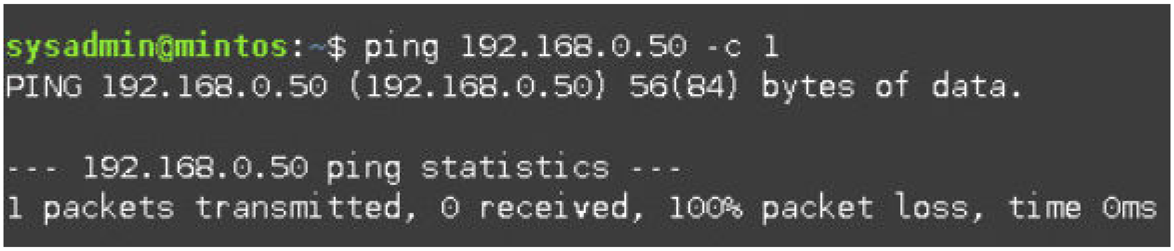


```
sysadmin@mintos:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
pfSense.home.arpa        ether   00:50:56:99:47:bd   C                     ens192
```

When a query for an address is made, the result is stored in the device's *ARP* table. Entries in the ARP table are cached for a limited amount of time and then timed out.

3. To refresh the *ARP* table, ping the *WinOS* computer with the command:

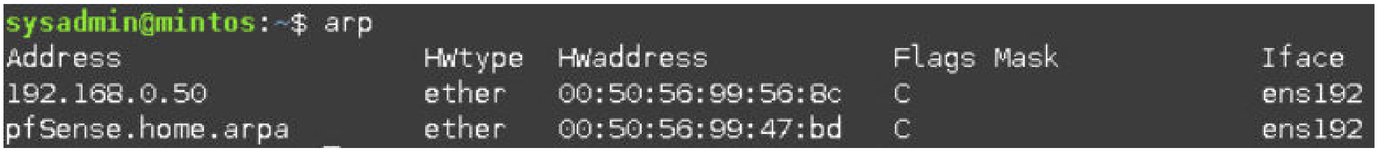
```
ping 192.168.0.50 –c 1
```



```
sysadmin@mintos:~$ ping 192.168.0.50 -c 1
PING 192.168.0.50 (192.168.0.50) 56(84) bytes of data.

--- 192.168.0.50 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

4. You will not receive a reply since the *Windows Firewall* is blocking pings. But, the *ping* will produce an *ARP* reply with the WinOS Mac Address. Run the **arp** command again, and you should see the *WinOS* host in the list.

```
arp
```



```
sysadmin@mintos:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.50             ether   00:50:56:99:56:8c   C                     ens192
pfSense.home.arpa        ether   00:50:56:99:47:bd   C                     ens192
```

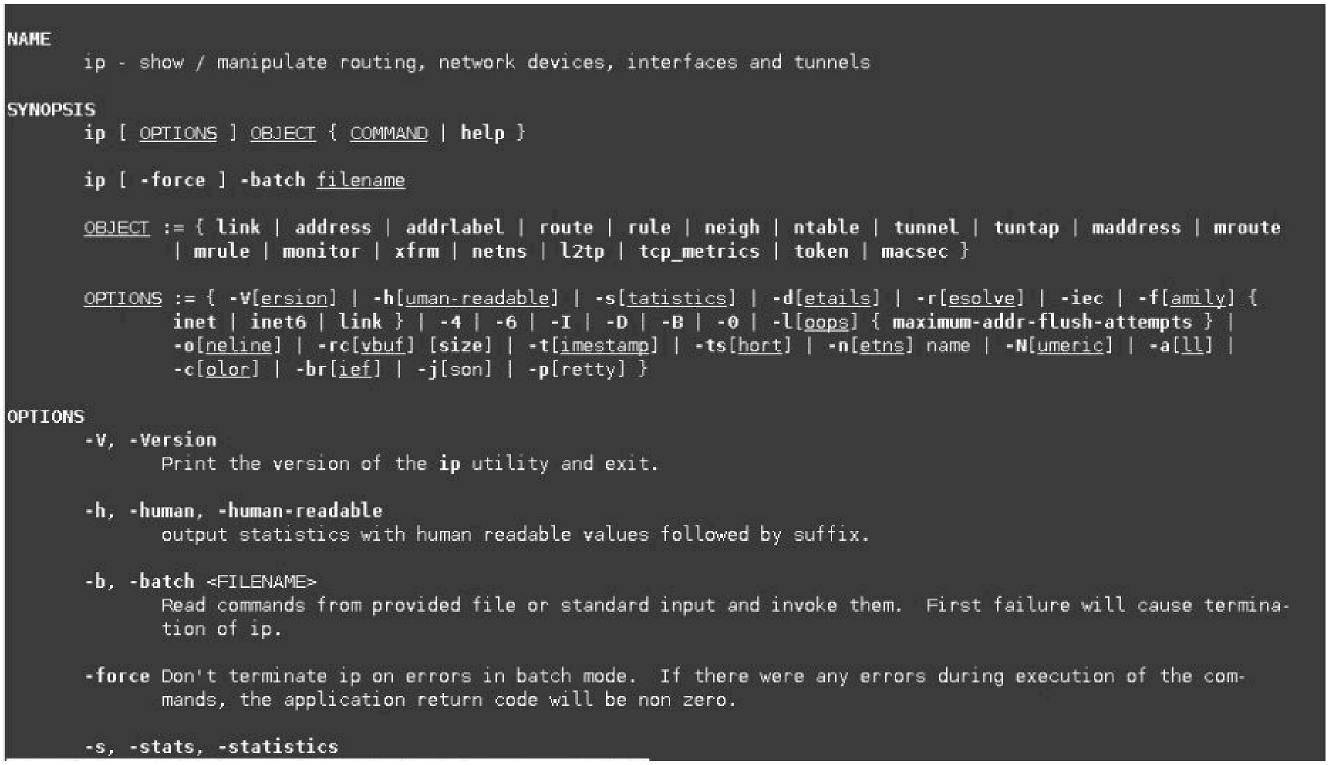
5. Leave the *MintOS* machine open and continue to the next task.

5.2 ip neighbor Command

The **ip** command suite, which includes the *ip neighbor* command, has replaced the deprecated *arp* command, which is part of the **Net-tools** package

1. In the terminal window, type the following command to show the manual page for *ip neigh*:

```
man ip neigh
```

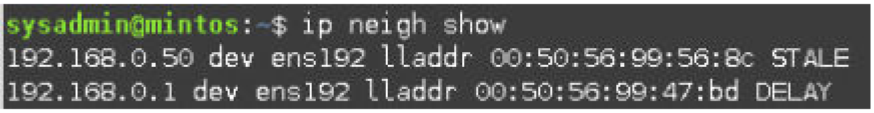


```
NAME
       ip - show / manipulate routing, network devices, interfaces and tunnels

SYNOPSIS
       ip [ OPTIONS ] OBJECT { COMMAND | help }

       ip [ -force ] -batch filename

       OBJECT := { link | address | addrlabel | route | rule | neigh | ntable | tunnel | tuntap | maddress | mroute
               | mrule | monitor | xfrm | netns | l2tp | tcp_metrics | token | macsec }

       OPTIONS := { -V[ersion] | -h[uman-readable] | -s[tatistics] | -d[etails] | -r[esolve] | -iec | -f[amily] {
               inet | inet6 | link } | -4 | -6 | -I | -D | -B | -0 | -l[oops] { maximum-addr-flush-attempts } |
               -o[neline] | -rc[vbuf] [size] | -t[imestamp] | -ts[hort] | -n[etns] name | -N[umeric] | -a[ll] |
               -c[olor] | -br[ief] | -j[son] | -p[retty] }

OPTIONS
       -V, -Version
              Print the version of the ip utility and exit.

       -h, -human, -human-readable
              output statistics with human readable values followed by suffix.

       -b, -batch <FILENAME>
              Read commands from provided file or standard input and invoke them.  First failure will cause termina-
              tion of ip.

       -force Don't terminate ip on errors in batch mode.  If there were any errors during execution of the com-
              mands, the application return code will be non zero.

       -s, -stats, -statistics
```

Type q to exit the *ip neigh* manual page.

2. To display the *ARP* table for all of the interfaces on the host, type the following command:

```
ip neigh show
```

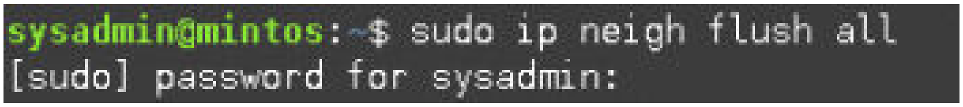


```
sysadmin@mintos:~$ ip neigh show
192.168.0.50 dev ens192 lladdr 00:50:56:99:56:8c STALE
192.168.0.1 dev ens192 lladdr 00:50:56:99:47:bd DELAY
```

3. To clear all the dynamic entries in the ARP table, type the following command:

```
sudo ip neigh flush all
```

If asked for the **[sudo] password**, type: `NDGlabpass123!`

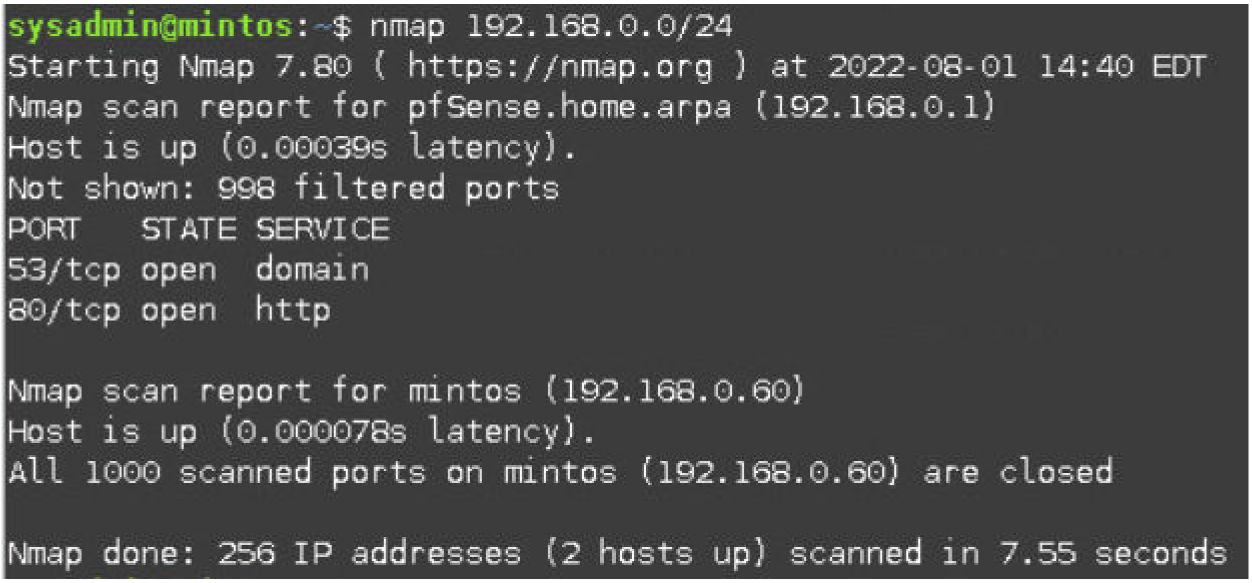


```
sysadmin@mintos:~$ sudo ip neigh flush all
[sudo] password for sysadmin:
```

If the command executed correctly, you will not get any reply message.

6. Use *nmap* to send an ARP Request to all computers on the 192.168.0.0 network by typing the following command:

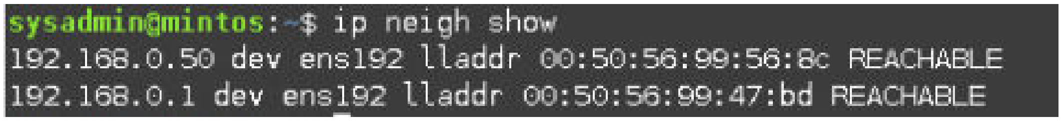
```
nmap 192.168.0.0/24
```



```
sysadmin@mintos:~$ nmap 192.168.0.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-08-01 14:40 EDT
Nmap scan report for pfSense.home.arpa (192.168.0.1)
Host is up (0.00039s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
53/tcp open  domain
80/tcp open  http

Nmap scan report for mintos (192.168.0.60)
Host is up (0.000078s latency).
All 1000 scanned ports on mintos (192.168.0.60) are closed

Nmap done: 256 IP addresses (2 hosts up) scanned in 7.55 seconds
```

4. To check the table to see that the two entries are back, type the command:

```
ip neigh show
```



```
sysadmin@mintos:~$ ip neigh show
192.168.0.50 dev ens192 lladdr 00:50:56:99:56:8c REACHABLE
192.168.0.1 dev ens192 lladdr 00:50:56:99:47:bd REACHABLE
```

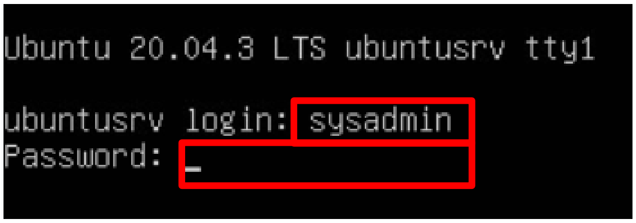
5. Close the *MintOS* terminal window by clicking the **X** in the upper-right corner.

6 Identifying Local User Account Information

A commonly overlooked vector for hackers to exploit is user accounts that are neglected, insecure, unknown, and even forgotten. In many cases, malware can be injected into a Linux system that creates user accounts that allows for bad actors to exploit. It's the task of the security analyst to monitor the user database and check to see if unauthorized users are logged in to the Linux system.

6.1 Using whoami to Identify the Current User

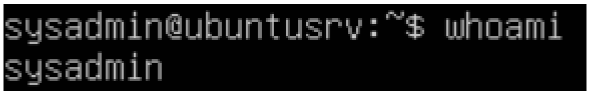
1. Change the focus on the **UbuntuSRV** and log in as `sysadmin` with the password: `NDGlabpass123!`



```
Ubuntu 20.04.3 LTS ubuntusrv tty1

ubuntusrv login: sysadmin
Password: _
```

2. The *whoami* command is used to display the currently logged-in user on the local Linux host. In the command prompt window, type the following command:

```
whoami
```



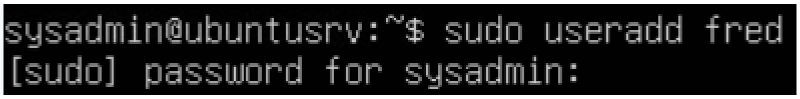
```
sysadmin@ubuntusrv:~$ whoami
sysadmin
```

3. Leave the *UbuntuSRV* machine open and continue to the next task.

6.2 List Users in Linux Local Database

1. When a new Linux system is installed, the installation process will ask you to create a user account. Additional accounts can be created for users to access services and resources on the server. Add a user to the **UbuntuSRV** system by typing the following command, when asked for the **[sudo] password** type: `NDGlabpass123!`

```
sudo useradd fred
```

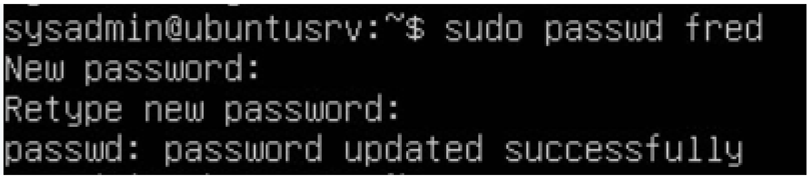


```
sysadmin@ubuntusrv:~$ sudo useradd fred
[sudo] password for sysadmin:
```

2. Assign a password to the user **fred** by typing the following command:

```
sudo passwd fred
```

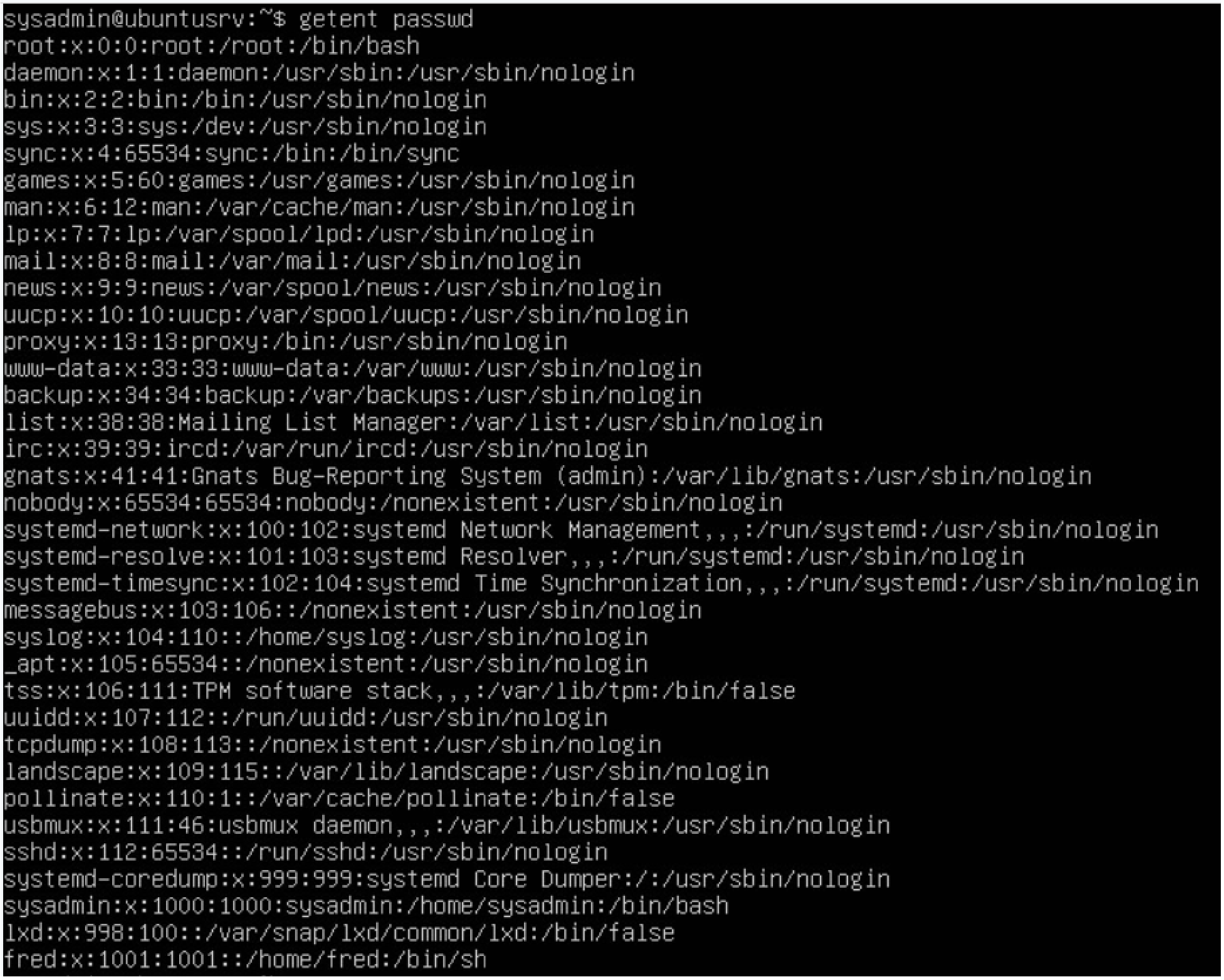
When asked for the *New Password*, type `Password1` and press **Enter**. When asked to *Retype New Password*, type `Password1` and press **Enter** again:



```
sysadmin@ubuntusrv:~$ sudo passwd fred
New password:
Retype new password:
passwd: password updated successfully
```

3. The list of users is kept in the **passwd** file, which is kept in the */etc* directory. This is a text file and can be opened and displayed with the **getent** command. At the command prompt, type the following command:

```
getent passwd
```



```
sysadmin@ubuntusrv:~$ getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sysadmin:x:1000:1000:sysadmin:/home/sysadmin:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
fred:x:1001:1001::/home/fred:/bin/sh
```

Each line in the file contains seven fields that are separated by a colon (:).

The seven fields are:

- User Name
- Encrypted Password. If there is an X, the encrypted password is stored in the /etc/shadow file.
- User ID (UID)
- Group ID (GID)
- User's Full Name
- User's Home Directory
- Login Shell, which shell this user will run, for example, bash or csh

There are two types of users in a Linux system, namely **System** and **Normal**.

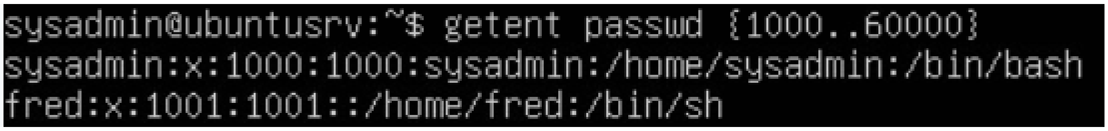
"System users are created when installing the OS and new packages. In some cases, you can create a system user that will be used by some applications.

Normal users are the users created by the root or another user with sudo privileges. Usually, a normal user has a real login shell and a home directory."

https://linuxize.com/post/how-to-list-users-in-linux/

4. Each user is assigned an ID number (**UID**). When a new user is created, a **UID** can be specified. If a **UID** is not entered, then the **UID** will be selected from a range of numbers from the */etc/login.defs* file. By default, the range of numbers used for normal users is **1000 – 60000,** and system users are assigned **UIDs** in the range of **100 – 999**. Using the default ranges, filter the list to only show the normal users by typing the following command:

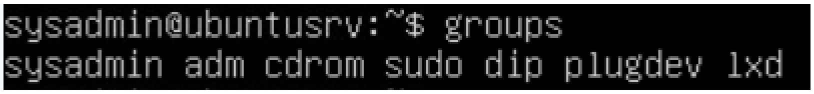
```
getent passwd {1000..60000}
```



```
sysadmin@ubuntusrv:~$ getent passwd {1000..60000}
sysadmin:x:1000:1000:sysadmin:/home/sysadmin:/bin/bash
fred:x:1001:1001::/home/fred:/bin/sh
```

5. It's also important to check the groups that the user belongs to. To see which groups the currently logged-in user belongs to, type the command:

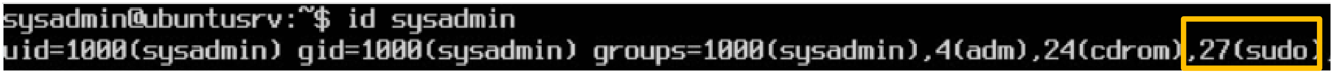
```
groups
```



```
sysadmin@ubuntusrv:~$ groups
sysadmin adm cdrom sudo dip plugdev lxd
```

6. The *id* command will display information about a specific user and the groups that the user belongs to. Type the following command to display the information for the sysadmin:

```
id sysadmin
```



```
sysadmin@ubuntusrv:~$ id sysadmin
uid=1000(sysadmin) gid=1000(sysadmin) groups=1000(sysadmin),4(adm),24(cdrom),27(sudo),
```

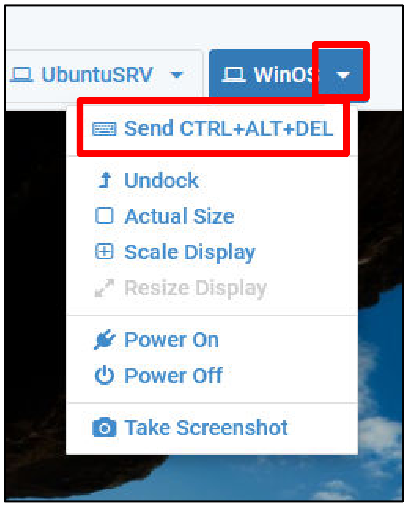
Notice the user is a member of the *sudo* group. Being a member of this group allows the user to execute elevated commands as the root user. This is the group that only system administrator accounts should be a member of.

6.3 Identify Users Logged In Remotely Using the who Command

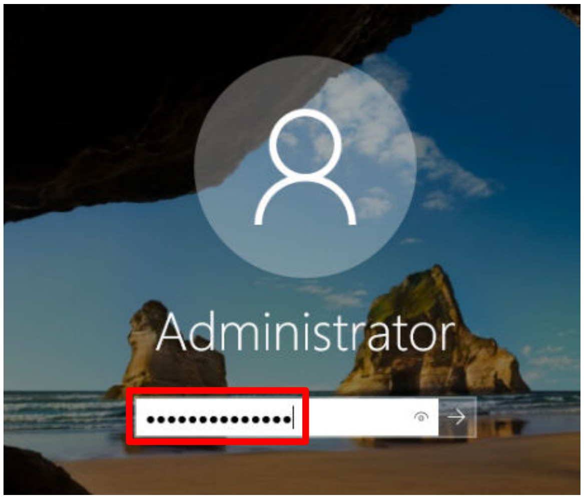
Linux and *Unix* have two applications/services that allow a user to log in to the computer remotely. The first is *Telnet*, which has been a part of the *TCP/IP* suite since the beginning. Even though the service is still functional, it is NOT secure. All transfers of data, including user names and passwords, are sent in plain text, with no encryption. The second client/server remote access service is *SSH* (Secure Shell) which is considered secure because it encrypts data transfers and uses Public Key Encryption for authentication.

However, if a Linux computer has already been compromised and a user account with **sudo** privilege has been created, then a bad actor can use *SSH* to log in to a computer remotely and wreak havoc. A security analyst needs to be able to monitor the user accounts that have logged in remotely and determine from which hosts they are connected.

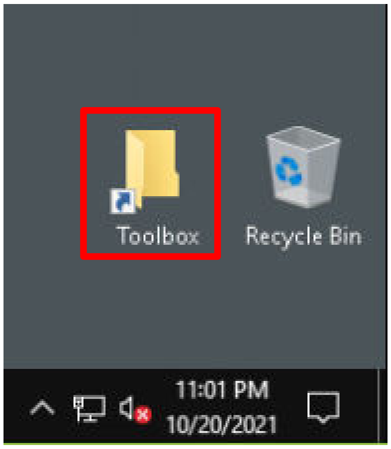
1. Change the focus to the **WinOS** host.
2. Bring up the login window by sending a Ctrl + Alt + Delete. To do this, click the **WinOS** dropdown menu and click **Send CTRL+ALT+DEL**.



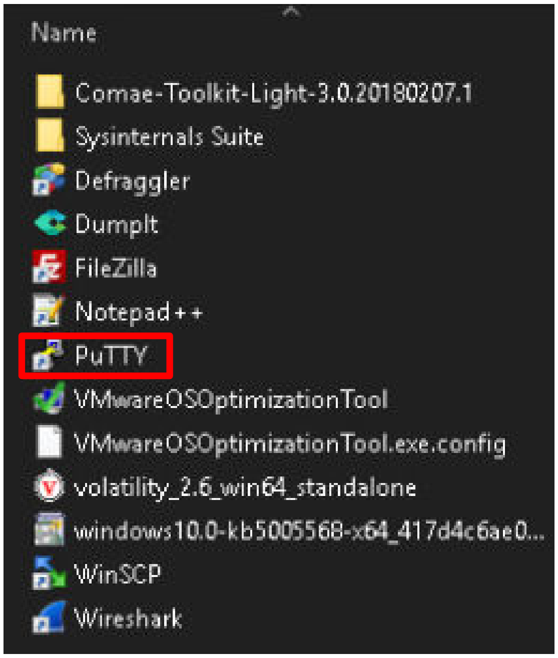
3. Log in as *Administrator* using the password: `NDGlabpass123!`



4. Double-click to open the **Toolbox** folder on the desktop.



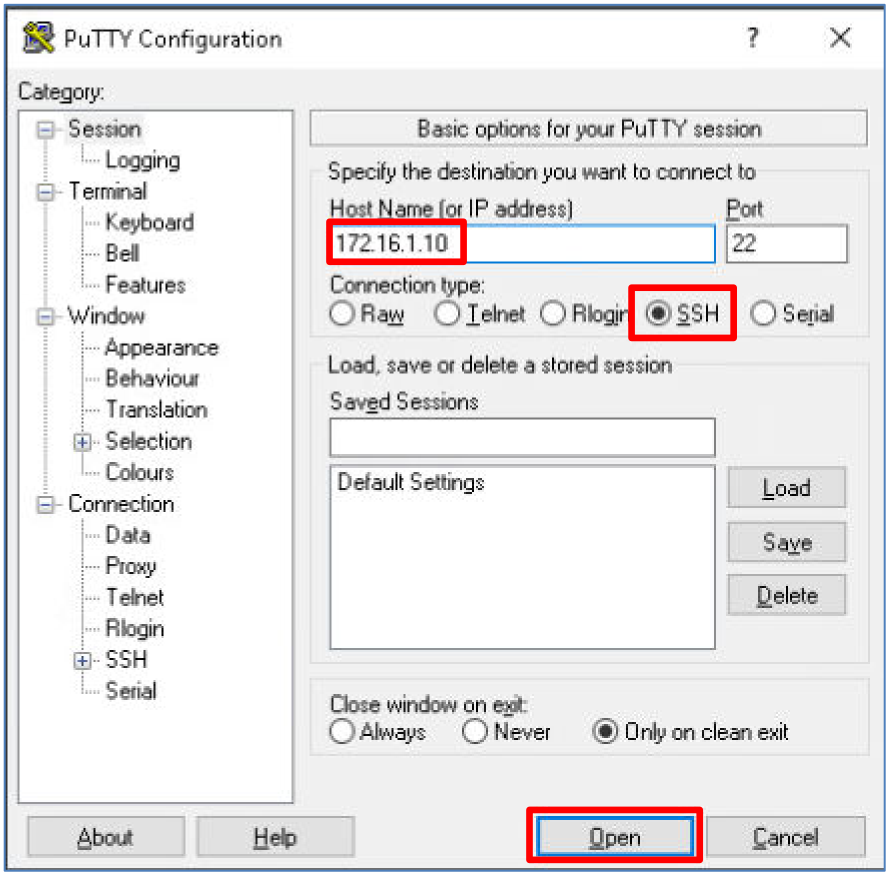
5. Open the **PuTTY** application by double-clicking on the file.



Name

- Comae-Toolkit-Light-3.0.20180207.1
- Sysinternals Suite
- Defraggler
- DumpIt
- FileZilla
- Notepad++
- PuTTY
- VMwareOSOptimizationTool
- VMwareOSOptimizationTool.exe.config
- volatility_2.6_win64_standalone
- windows10.0-kb5005568-x64_417d4c6ae0...
- WinSCP
- Wireshark

6. In the *PuTTY* configuration window, type the IP address of the *UbuntuSRV* computer (`172.16.1.10`), make sure the **SSH** radio button is selected, and click the **Open** button.



Specify the destination you want to connect to

Host Name (or IP address): 172.16.1.10

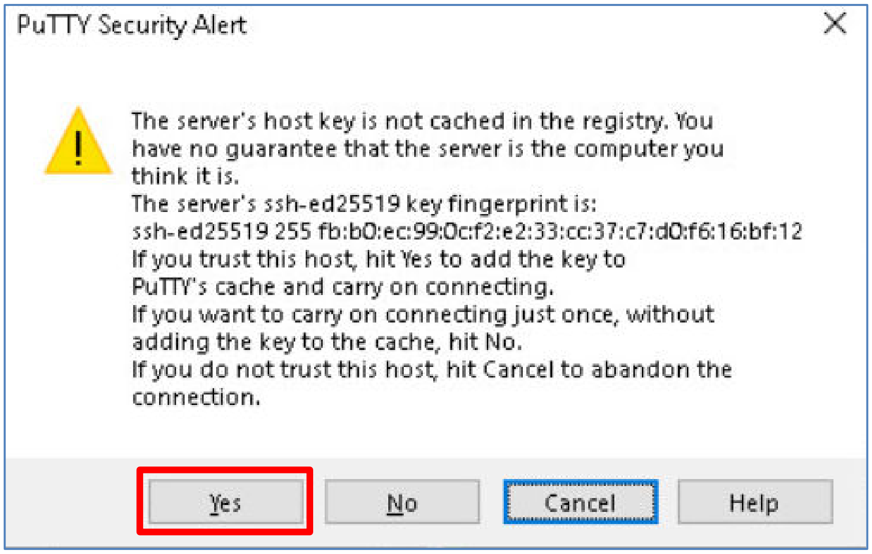
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: Default Settings

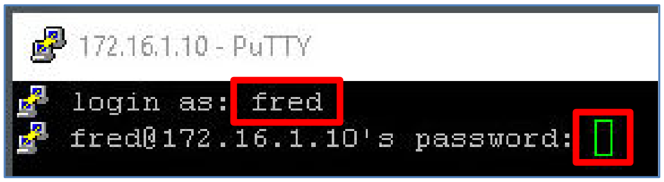
Close window on exit: ○ Always ○ Never ● Only on clean exit

7. When the *PuTTY Security Alert* window pops up, click the **Yes** button.



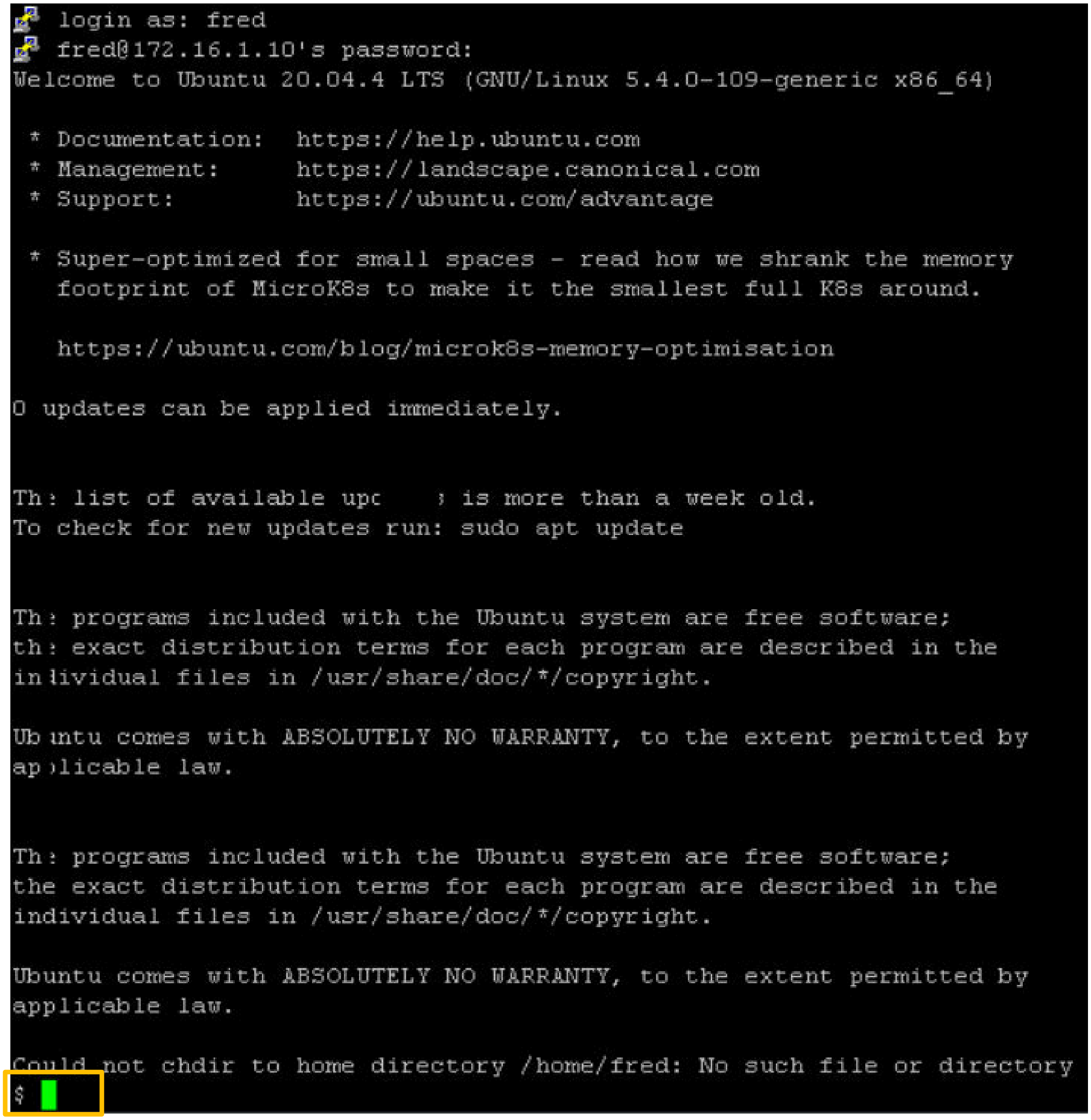
The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 fb:b0:ec:99:0c:f2:e2:33:cc:37:c7:d0:f6:16:bf:12
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

8. At the *login as* prompt, type the username `fred`, then type the password: `Password1`



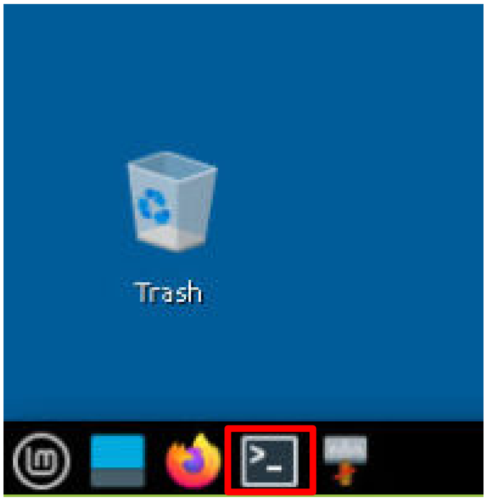
```
login as: fred
fred@172.16.1.10's password:
```

After successfully logging in, you should see the screen below and the $ prompt.



```
login as: fred
fred@172.16.1.10's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

0 updates can be applied immediately.


Th: list of available upc    : is more than a week old.
To check for new updates run: sudo apt update


Th: programs included with the Ubuntu system are free software;
th: exact distribution terms for each program are described in the
in lividual files in /usr/share/doc/*/copyright.

Ub intu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
ap )licable law.


Th: programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Could not chdir to home directory /home/fred: No such file or directory
$
```

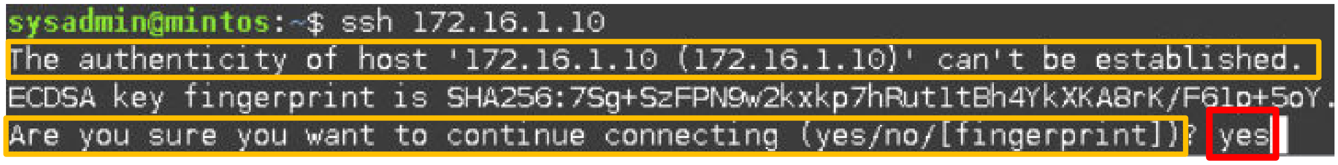
9. Click on the **MintOS** tab to open the Linux Mint computer
10. Click on the **Terminal** icon in the taskbar at the bottom of the screen.



11. Type the following command to log in to the *UbuntSRV* computer.

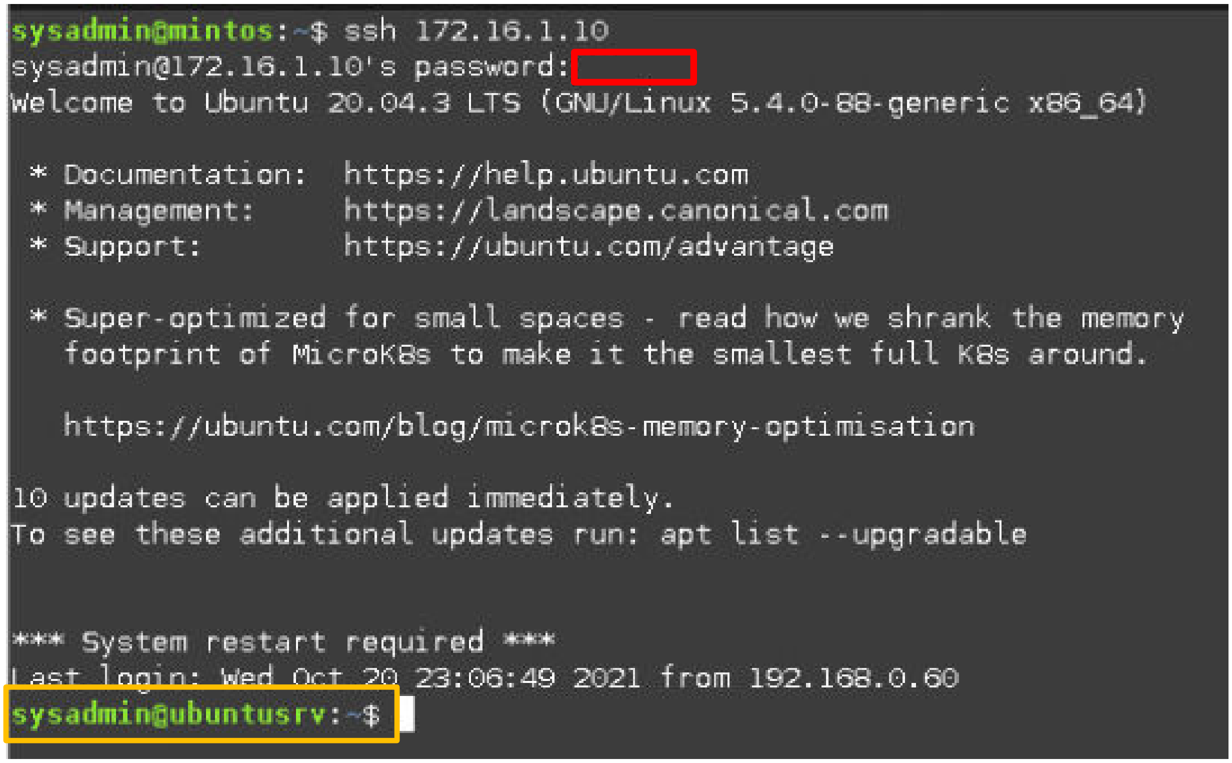
```
ssh 172.16.1.10
```

If you see the message about the authenticity of the host can't be established, and do you want to continue, type **yes** and press **Enter.**



```
sysadmin@mintos:~$ ssh 172.16.1.10
The authenticity of host '172.16.1.10 (172.16.1.10)' can't be established.
ECDSA key fingerprint is SHA256:7Sg+SzFPN9w2kxkp7hRut1tBh4YkXKA8rK/F61p+5oY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

12. At the **sysadmin@172.16.1.10** password prompt, type: `NDGlabpass123!`



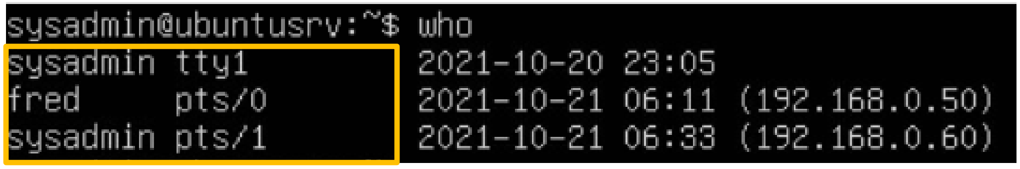
```
sysadmin@mintos:~$ ssh 172.16.1.10
sysadmin@172.16.1.10's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Wed Oct 20 23:06:49 2021 from 192.168.0.60
sysadmin@ubuntusrv:~$
```

Notice the prompt has changed from **sysadmin@mintos** to **sysadmin@ubuntusrv,** indicating that the user has logged in remotely to the *UbuntuSRV* computer.

In Linux, *SSH* does not prompt for a user name. If no username is provided using the **–l <*login_name*>** option, then the user that is logged into the computer that SSH was run from will be used. If there is a user with the same name on the remote computer, *SSH* will ask for the password, otherwise the login will terminate.

13. Change focus to the **UbuntuSRV** computer.
14. At the command prompt, type the following command to see the local and remote users that are currently logged in.

```
who
```



```
sysadmin@ubuntusrv:~$ who
sysadmin tty1         2021-10-20 23:05
fred     pts/0        2021-10-21 06:11 (192.168.0.50)
sysadmin pts/1        2021-10-21 06:33 (192.168.0.60)
```

- The first field shows the user name.
- The second field shows how the user connected; tty means the user is connected locally, and pts means the user connected remotely.
- The third column is the date the user logged in.
- The fourth column is the time the user logged in.
- The fifth column shows the IP address of the host the remote user logged into.

The security analyst can now determine if an unauthorized user has logged in and from where.

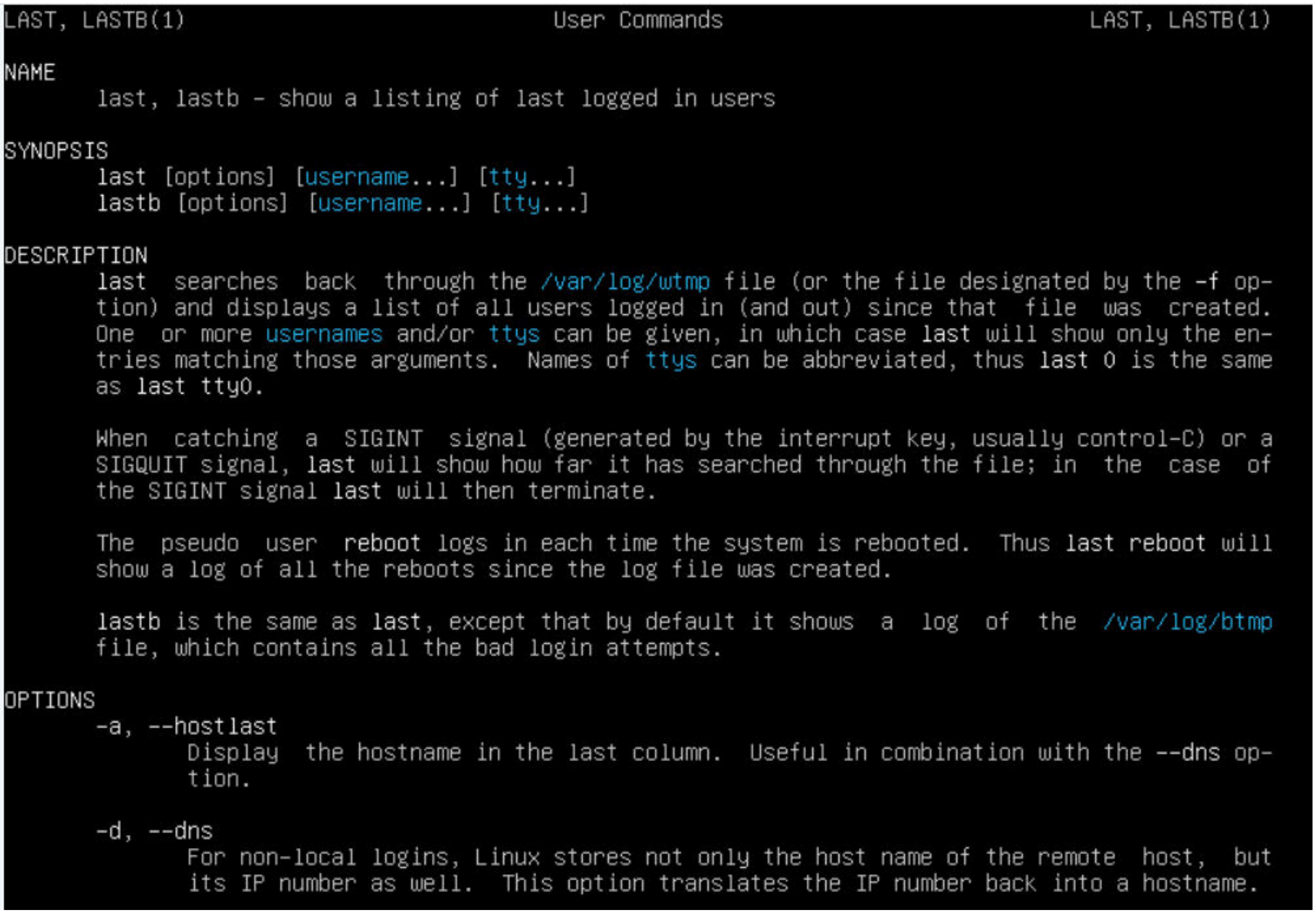
15. Leave the *UbuntuSRV* computer open for the next task.

6.4 Viewing Login History Using last

A large part of the security analyst's mission is to use system logs to identify security holes, vulnerabilities, and anomalies. One of the commonly used tools that the security analyst uses is the *last* command which can view the **/var/log/wtmp** log, which displays the list of users, both local and remote, that have logged into a Linux system. The *last* command has many options that can be used to show different views of the logs, such as the history of a particular user or the IP addresses of the computers that have been used for logging in remotely.

1. Type the following command to view the manual entry for the last command.

```
man last
```

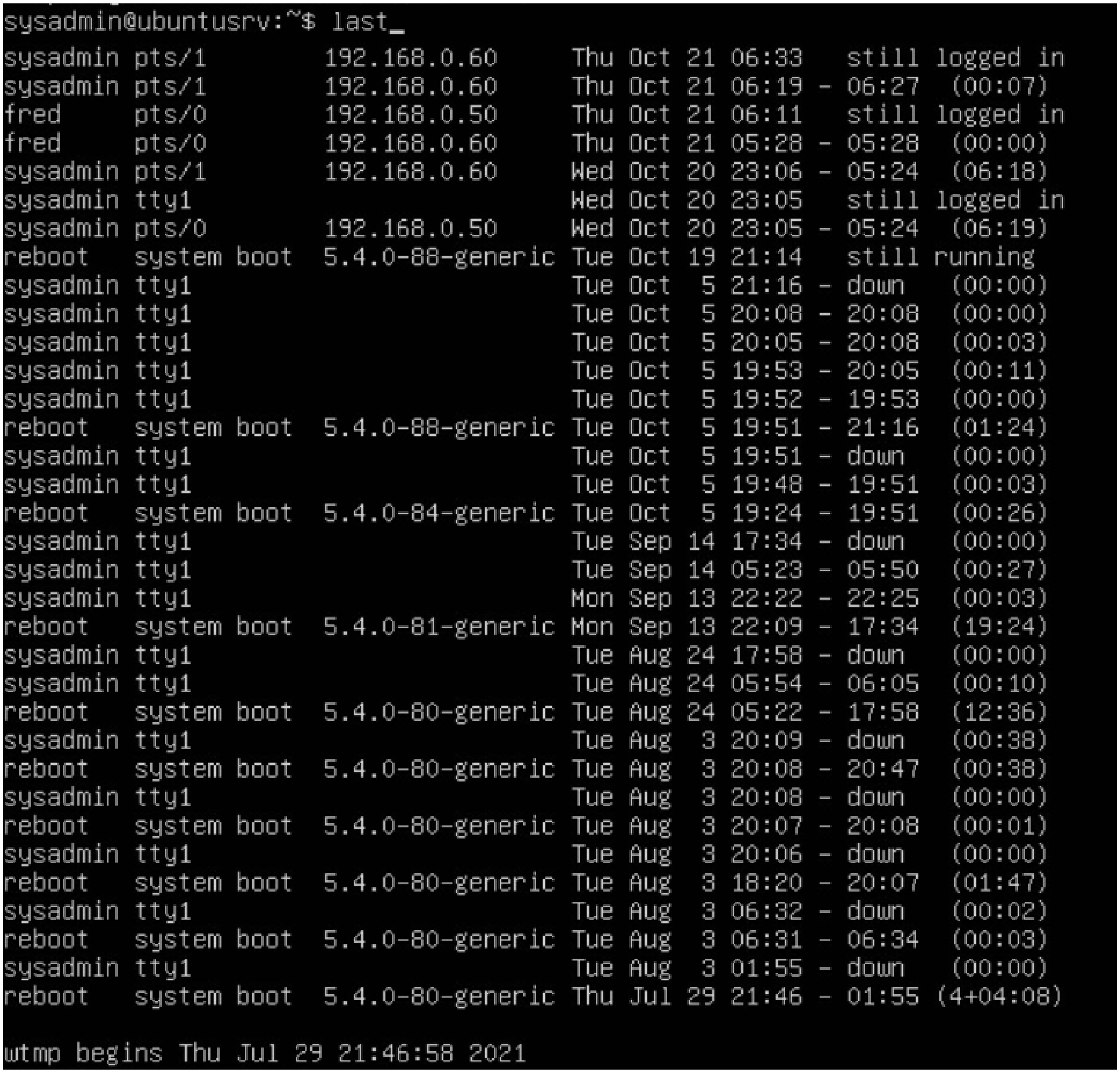


```
LAST, LASTB(1)                     User Commands                    LAST, LASTB(1)

NAME
       last, lastb - show a listing of last logged in users

SYNOPSIS
       last [options] [username...] [tty...]
       lastb [options] [username...] [tty...]

DESCRIPTION
       last  searches  back  through the /var/log/wtmp file (or the file designated by the -f op‐
       tion) and displays a list of all users logged in (and out) since that  file  was  created.
       One  or more usernames and/or ttys can be given, in which case last will show only the en‐
       tries matching those arguments.  Names of ttys can be abbreviated, thus last 0 is the same
       as last tty0.

       When  catching  a  SIGINT  signal (generated by the interrupt key, usually control-C) or a
       SIGQUIT signal, last will show how far it has searched through the file; in  the  case  of
       the SIGINT signal last will then terminate.

       The  pseudo  user  reboot logs in each time the system is rebooted.  Thus last reboot will
       show a log of all the reboots since the log file was created.

       lastb is the same as last, except that by default it shows  a  log  of  the  /var/log/btmp
       file, which contains all the bad login attempts.

OPTIONS
       -a, --hostlast
              Display  the hostname in the last column.  Useful in combination with the --dns op‐
              tion.

       -d, --dns
              For non-local logins, Linux stores not only the host name of the remote  host,  but
              its IP number as well.  This option translates the IP number back into a hostname.
```

Type q to exit the *last* manual page.

2. Type the following command to show the default display from the /var/log/wtmp log file.

```
last
```

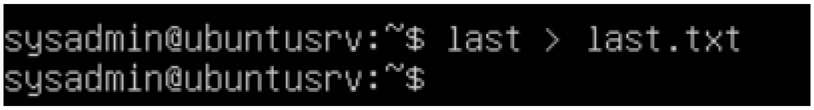


```
sysadmin@ubuntusrv:~$ last_
sysadmin pts/1        192.168.0.60     Thu Oct 21 06:33   still logged in
sysadmin pts/1        192.168.0.60     Thu Oct 21 06:19 - 06:27  (00:07)
fred     pts/0        192.168.0.50     Thu Oct 21 06:11   still logged in
fred     pts/0        192.168.0.60     Thu Oct 21 05:28 - 05:28  (00:00)
sysadmin pts/1        192.168.0.60     Wed Oct 20 23:06 - 05:24  (06:18)
sysadmin tty1                          Wed Oct 20 23:05   still logged in
sysadmin pts/0        192.168.0.50     Wed Oct 20 23:05 - 05:24  (06:19)
reboot   system boot  5.4.0-88-generic Tue Oct 19 21:14   still running
sysadmin tty1                          Tue Oct  5 21:16 - down   (00:00)
sysadmin tty1                          Tue Oct  5 20:08 - 20:08  (00:00)
sysadmin tty1                          Tue Oct  5 20:05 - 20:08  (00:03)
sysadmin tty1                          Tue Oct  5 19:53 - 20:05  (00:11)
sysadmin tty1                          Tue Oct  5 19:52 - 19:53  (00:00)
reboot   system boot  5.4.0-88-generic Tue Oct  5 19:51 - 21:16  (01:24)
sysadmin tty1                          Tue Oct  5 19:51 - down   (00:00)
sysadmin tty1                          Tue Oct  5 19:48 - 19:51  (00:03)
reboot   system boot  5.4.0-84-generic Tue Oct  5 19:24 - 19:51  (00:26)
sysadmin tty1                          Tue Sep 14 17:34 - down   (00:00)
sysadmin tty1                          Tue Sep 14 05:23 - 05:50  (00:27)
sysadmin tty1                          Mon Sep 13 22:22 - 22:25  (00:03)
reboot   system boot  5.4.0-81-generic Mon Sep 13 22:09 - 17:34  (19:24)
sysadmin tty1                          Tue Aug 24 17:58 - down   (00:00)
sysadmin tty1                          Tue Aug 24 05:54 - 06:05  (00:10)
reboot   system boot  5.4.0-80-generic Tue Aug 24 05:22 - 17:58  (12:36)
sysadmin tty1                          Tue Aug  3 20:09 - down   (00:38)
reboot   system boot  5.4.0-80-generic Tue Aug  3 20:08 - 20:47  (00:38)
sysadmin tty1                          Tue Aug  3 20:08 - down   (00:00)
reboot   system boot  5.4.0-80-generic Tue Aug  3 20:07 - 20:08  (00:01)
sysadmin tty1                          Tue Aug  3 20:06 - down   (00:00)
reboot   system boot  5.4.0-80-generic Tue Aug  3 18:20 - 20:07  (01:47)
sysadmin tty1                          Tue Aug  3 06:32 - down   (00:02)
reboot   system boot  5.4.0-80-generic Tue Aug  3 06:31 - 06:34  (00:03)
sysadmin tty1                          Tue Aug  3 01:55 - down   (00:00)
reboot   system boot  5.4.0-80-generic Thu Jul 29 21:46 - 01:55 (4+04:08)

wtmp begins Thu Jul 29 21:46:58 2021
```

The listing shows the user name, whether they logged in locally (**tty**) or remotely (**pts**), the IP address of the host if they logged in remotely, the day, date, and time the user logged in and out and how long the user was logged in, or even if they are still logged in at the time the list was generated. Additionally, the list shows the date and time the Linux computer was rebooted or shutdown.

3. To save the output into a file for further analysis and documentation, type the following to run the last command and write the output to a file rather than the display.

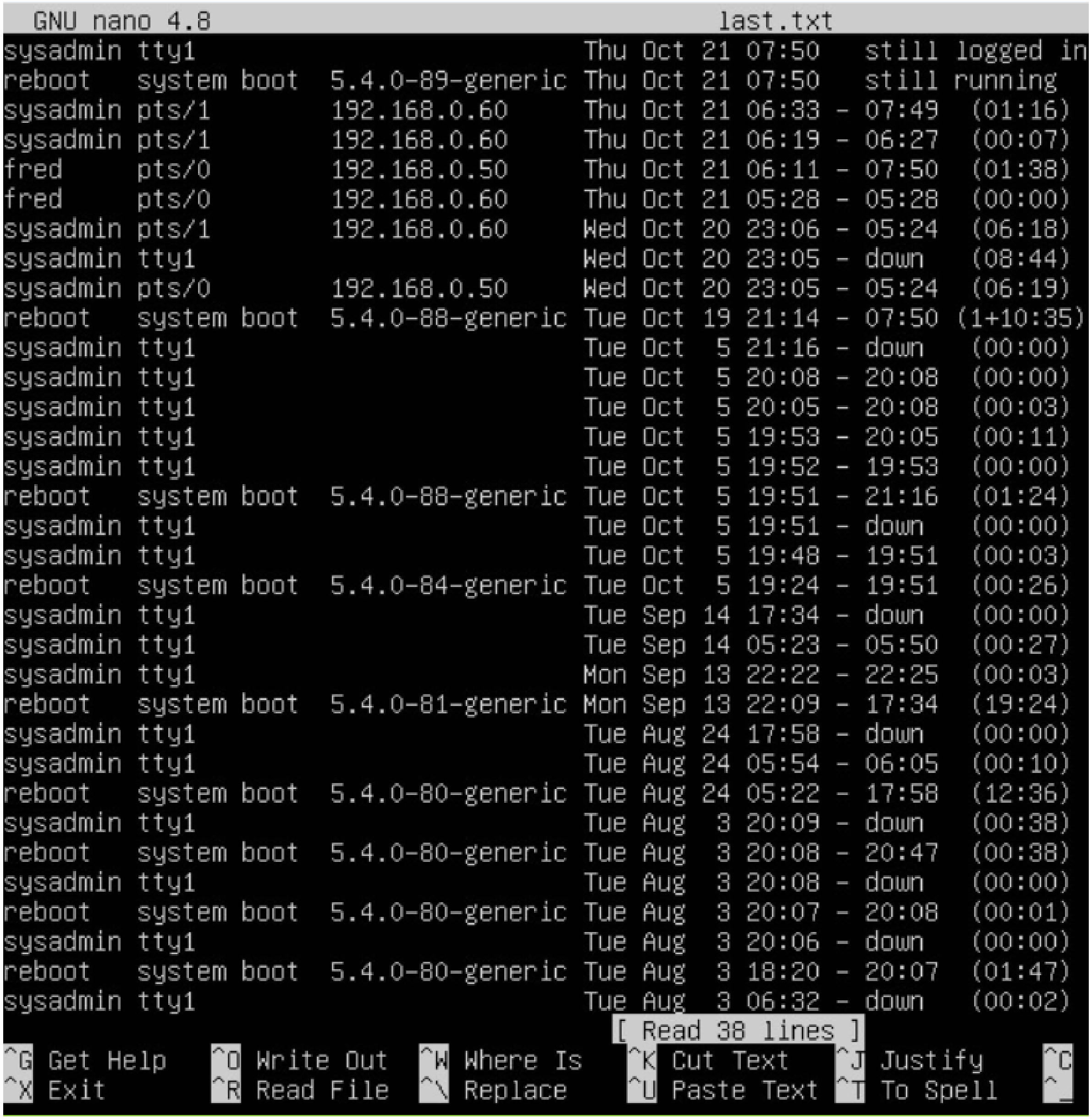
```
last > last.txt
```



```
sysadmin@ubuntusrv:~$ last > last.txt
sysadmin@ubuntusrv:~$
```

4. Open the text file in the *nano* text editor by typing the following command:

```
nano last.txt
```



```
  GNU nano 4.8                        last.txt
sysadmin tty1                          Thu Oct 21 07:50   still logged in
reboot   system boot  5.4.0-89-generic Thu Oct 21 07:50   still running
sysadmin pts/1        192.168.0.60     Thu Oct 21 06:33 - 07:49  (01:16)
sysadmin pts/1        192.168.0.60     Thu Oct 21 06:19 - 06:27  (00:07)
fred     pts/0        192.168.0.50     Thu Oct 21 06:11 - 07:50  (01:38)
fred     pts/0        192.168.0.60     Thu Oct 21 05:28 - 05:28  (00:00)
sysadmin pts/1        192.168.0.60     Wed Oct 20 23:06 - 05:24  (06:18)
sysadmin tty1                          Wed Oct 20 23:05 - down   (08:44)
sysadmin pts/0        192.168.0.50     Wed Oct 20 23:05 - 05:24  (06:19)
reboot   system boot  5.4.0-88-generic Tue Oct 19 21:14 - 07:50 (1+10:35)
sysadmin tty1                          Tue Oct  5 21:16 - down   (00:00)
sysadmin tty1                          Tue Oct  5 20:08 - 20:08  (00:00)
sysadmin tty1                          Tue Oct  5 20:05 - 20:08  (00:03)
sysadmin tty1                          Tue Oct  5 19:53 - 20:05  (00:11)
sysadmin tty1                          Tue Oct  5 19:52 - 19:53  (00:00)
reboot   system boot  5.4.0-88-generic Tue Oct  5 19:51 - 21:16  (01:24)
sysadmin tty1                          Tue Oct  5 19:51 - down   (00:00)
sysadmin tty1                          Tue Oct  5 19:48 - 19:51  (00:03)
reboot   system boot  5.4.0-84-generic Tue Oct  5 19:24 - 19:51  (00:26)
sysadmin tty1                          Tue Sep 14 17:34 - down   (00:00)
sysadmin tty1                          Tue Sep 14 05:23 - 05:50  (00:27)
sysadmin tty1                          Mon Sep 13 22:22 - 22:25  (00:03)
reboot   system boot  5.4.0-81-generic Mon Sep 13 22:09 - 17:34  (19:24)
sysadmin tty1                          Tue Aug 24 17:58 - down   (00:00)
sysadmin tty1                          Tue Aug 24 05:54 - 06:05  (00:10)
reboot   system boot  5.4.0-80-generic Tue Aug 24 05:22 - 17:58  (12:36)
sysadmin tty1                          Tue Aug  3 20:09 - down   (00:38)
reboot   system boot  5.4.0-80-generic Tue Aug  3 20:08 - 20:47  (00:38)
sysadmin tty1                          Tue Aug  3 20:08 - down   (00:00)
reboot   system boot  5.4.0-80-generic Tue Aug  3 20:07 - 20:08  (00:01)
sysadmin tty1                          Tue Aug  3 20:06 - down   (00:00)
reboot   system boot  5.4.0-80-generic Tue Aug  3 18:20 - 20:07  (01:47)
sysadmin tty1                          Tue Aug  3 06:32 - down   (00:02)
                                    [ Read 38 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_
```

5. Press **CTRL-X** to exit from the *nano* editor.
6. The lab is now complete; you may now end the reservation.