
Assignment No: 1

---

1. Explore and Study of TCP/IP utilities and Network Commands on Linux.

a) Ping
b) ipconfig / ifconfig
c) Hostname
d) Whois
e) Netstat
f) Route
g) Tracert/Traceroute/Tracepath
h) NSlookup
i) Arp
j) Finger
k) Port Scan / nmap

1. Ping command:

The ping command is a Command Prompt command used to test the ability of the source computer to reach a specified destination computer. The ping command is usually used as a simple way to verify that a computer can communicate over the network with another computer or network device.

The ping command operates by sending *Internet Control Message Protocol (ICMP)* *Echo Request* messages to the destination computer and waiting for a response.

How many of those responses are returned, and how long it takes for them to return, are the two major pieces of information that the ping command provides.

For example, you might find that there are no responses when pinging a network printer, only to find out that the printer is offline and its cable needs replaced. Or maybe you need to ping a router to verify that your computer can connect to it, to eliminate it as a possible cause for a networking issue.

Ping Command Syntax

ping [**-t**] [**-a**] [**-n** *count*] [**-l** *size*] [**-f**] [**-i** *TTL*] [**-v** *TOS*] [**-r** *count*] [**-s** *count*] [**-w** *timeout*] [**-R**] [**-S** *srcaddr*] [**-p**] [**-4**] [**-6**] *target* [**/?**]

Tip: See How to Read Command Syntax if you're not sure how to interpret the ping command syntax as it's described above or in the table below.

-t	Using this option will ping the *target* until you force it to stop by using Ctrl-C.
-a	This ping command option will resolve, if possible, the hostnameof an IP address *target*.
-n *count*	This option sets the number of ICMP Echo Requests to send, from 1 to 4294967295. The ping command will send 4 by default if **-n** isn't used.
-l *size*	Use this option to set the size, in bytes, of the echo request packet from 32 to 65,527. The ping command will send a 32-byte echo request if you don't use the **-l** option.

-f	Use this ping command option to prevent ICMP Echo Requests from being fragmented by routers between you and the *target*. The **-f** option is most often used to troubleshoot Path Maximum Transmission Unit (PMTU) issues.
-i *TTL*	This option sets the Time to Live (TTL) value, the maximum of which is 255.
-v *TOS*	This option allows you to set a Type of Service (TOS) value. Beginning in Windows 7, this option no longer functions but still exists for compatibility reasons.
-r *count*	Use this ping command option to specify the number of hopsbetween your computer and the *target* computer or device that you'd like to be recorded and displayed. The maximum value for *count* is 9, so use the tracert command instead if you're interested in viewing all the hops between two devices.
-s *count*	Use this option to report the time, in Internet Timestamp format, that each echo request is received and echo reply is sent. The maximum value for *count* is 4, meaning that only the first four hops can be time stamped.
-w *timeout*	Specifying a *timeout* value when executing the ping command adjusts the amount of time, in milliseconds, that ping waits for each reply. If you don't use the **-w** option, the default timeout value of 4000 is used, which is 4 seconds.
-R	This option tells the ping command to trace the round trip path.
-S *srcaddr*	Use this option to specify the source address.
-p	Use this switch to ping a *Hyper-V Network Virtualization* provider address.
-4	This forces the ping command to use IPv4 only but is only necessary if *target* is a hostname and not an IP address.
-6	This forces the ping command to use IPv6 only but as with the **-4**option, is only necessary when pinging a hostname.
target	This is the destination you wish to ping, either an IP address or a hostname.
/?	Use the help switch with the ping command to show detailed help about the command's several options.

Note: The **-f**, **-v**, **-r**, **-s**, **-j**, and **-k** options work when pinging IPv4 addresses only. The **-R** and **-S** options only work with IPv6.

Other less commonly used switches for the ping command exist including [**-j** *host-list*], [**-k** *host-list*], and [**-c** *compartment*]. Execute **ping /?** from the Command Prompt for more information on these options.

NAME : ifconfig - configure a network interface

SYNOPSIS

ifconfig [interface]

ifconfig interface [aftype] options | address ...

DESCRIPTION

Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed. If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

OPTIONS

TAG	DESCRIPTION
interface	**The name of the interface. This is usually a driver name followed by a unit number, for example eth0 for the first Ethernet interface.**
down	**This flag causes the driver for this interface to be shut down.**
[-]arp	**Enable or disable the use of the ARP protocol on this interface.**
[-]promisc	**Enable or disable the promiscuous mode of the interface. If selected, all packets on the network will be received by the interface.**
[-]allmulti	**Enable or disable all-multicast mode. If selected, all multicast packets on the network will be received by the interface.**
metric N	**This parameter sets the interface metric. It is not available under GNU/Linux.**
mtu N	**This parameter sets the Maximum Transfer Unit (MTU) of an interface.**
dstaddr addr	**Set the remote IP address for a point-to-point link (such as PPP). This keyword is now obsolete; use the pointopoint keyword instead.**
netmask addr	**Set the IP network mask for this interface. This value defaults to the usual class A, B or C network mask (as derived from the interface IP address), but it can be set to any value. .**

add addr/prefixlen	**Add an IPv6 address to an interface.**
del addr/prefixlen	**Remove an IPv6 address from an interface.**
tunnel ::aa.bb.cc.dd	**Create a new SIT (IPv6-in-IPv4) device, tunnelling to the given destination.**
irq addr	**Set the interrupt line used by this device. Not all devices can dynamically change their IRQ setting..**
io_addr addr	**Set the start address in I/O space for this device..**
mem_start addr	**TSet the start address for shared memory used by this device. Only a few devices need this..**
[-]broadcast [addr]	**If the address argument is given, set the protocol broadcast address for this interface. Otherwise, set (or clear) the IFF_BROADCAST flag for the interface..**
[-]pointopoint [addr]	**This keyword enables the point-to-point mode of an interface, meaning that it is a direct link between two machines with nobody else listening on it.**
hw class address	**Set the hardware address of this interface, if the device driver supports this operation. The keyword must be followed by the name of the hardware class and the printable ASCII equivalent of the hardware address. Hardware classes currently supported include ether (Ethernet), ax25 (AMPR AX.25), ARCnet and netrom (AMPR NET/ROM).**
multicast	**Set the multicast flag on the interface. This should not normally be needed as the drivers set the flag correctly themselves..**
address	**The IP address to be assigned to this interface.**
txqueuelen length	**Set the length of the transmit queue of the device. It is useful to set this to small values for slower devices with a high latency (modem links, ISDN) to prevent fast bulk transfers from disturbing interactive traffic like telnet too much**

HOSTNAME :

A hostname command is used to view a computer's hostname and domain name (DNS) (Domain Name Service), and to display or set a computer's hostname or domain name.
A hostname is a name that is given to a computer that attached to the network that uniquely identifies over a network and thus allows it to be accessed without using its IP address.
The basic syntax for the hostname command is:

hostname [options] [new_host_name]

If you run hostname command without any options, it will displays the current host name and domain name of your Linux system.

```
$ hostname
```

tecmint

```
[root@tecmint ~]# hostname
tecmint.com
[root@tecmint ~]#
```

Show Linux Hostname

If the host name can be resolved, you can display the network address(es) (IP address) of the host name with the -i flag and the -I option establishes all configured network interfaces and shows all network addresses of the host.

```
$ hostname -i

$ hostname -I
```

```
[root@tecmint ~]# hostname -i
192.168.0.1
[root@tecmint ~]# hostname -I
192.168.0.1 3a03:7b00::f13c:91ff:fedb:134560
[root@tecmint ~]#
```

WHOIS

whois is a client for the WHOIS directory service.

Description

whois searches for an object in a WHOIS database. WHOIS is a query and response protocol that is widely used for querying databases that store the registered users of an Internet resource, such as a domain name or an IP address block, but is also used for a wider range of other information.

Most modern versions of whois try to guess the right server to ask for the specified object. If no guess can be made, whois will connect to whois.networksolutions.comfor NIC handles or whois.arin.net for IPv4 addresses and network names.

whois syntax

```
whois [ -h HOST ] [ -p PORT ] [ -aCFHlLMmrRSVx ] [ -g SOURCE:FIRST-LAST ]

    [ -i ATTR ] [ -S SOURCE ] [ -T TYPE ] object

whois -t TYPE
```

```
whois -v TYPE

whois -q keyword
```

Options

-h HOST	Connect to WHOIS database host HOST.
-H	Suppress the display of legal disclaimers.
-p PORT	When connecting, connect to network port PORT.
--verbose	Operate verbosely.
--help	Display a help message, and exit.

Netstate :

netstat (network statistics) is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc. netstat is available on all Unix-like Operating Systems and also available on Windows OS as well. It is very useful in terms of network troubleshooting and performance measurement. netstat is one of the most basic network service debugging tools, telling you what ports are open and whether any programs are listening on ports.

This tool is very important and much useful for Linux network administrators as well as system administrators to monitor and troubleshoot their network related problems and determine network traffic performance.

You might also be interested in following article

1. Listing all the LISTENING Ports of TCP and UDP connections

Listing all ports (both TCP and UDP) using netstat -a option.

```
# netstat -a | more

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
tcp        0      0 *:sunrpc                    *:*
LISTEN
tcp        0     52 192.168.0.2:ssh             192.168.0.1:egs
ESTABLISHED
tcp        1      0 192.168.0.2:59292           www.gov.com:http
CLOSE_WAIT
tcp        0      0 localhost:smtp              *:*
LISTEN
tcp        0      0 *:59482                     *:*
LISTEN
udp        0      0 *:35036                     *:*
udp        0      0 *:npmp-local                *:*

Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     16972  /tmp/orbit-
root/linc-76b-0-6fa08790553d6
```

```
unix  2      [ ACC ]     STREAM     LISTENING     17149  /tmp/orbit-
root/linc-794-0-7058d584166d2
unix  2      [ ACC ]     STREAM     LISTENING     17161  /tmp/orbit-
root/linc-792-0-546fe905321cc
unix  2      [ ACC ]     STREAM     LISTENING     15938  /tmp/orbit-
root/linc-74b-0-415135cb6aeab
```

2. Listing TCP Ports connections

Listing only TCP (Transmission Control Protocol) port connections using netstat -at.

```
# netstat -at

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
tcp        0      0 *:ssh                       *:*
LISTEN
tcp        0      0 localhost:ipp               *:*
LISTEN
tcp        0      0 localhost:smtp              *:*
LISTEN
tcp        0     52 192.168.0.2:ssh             192.168.0.1:egs
ESTABLISHED
tcp        1      0 192.168.0.2:59292           www.gov.com:http
CLOSE_WAIT
```

3. Listing UDP Ports connections

Listing only UDP (User Datagram Protocol) port connections using netstat -au.

```
# netstat -au

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
udp        0      0 *:35036                     *:*
udp        0      0 *:npmp-local                *:*
udp        0      0 *:mdns                      *:*
```

4. Listing all LISTENING Connections

Listing all active listening ports connections with netstat -l.

```
# netstat -l
```

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
tcp        0      0 *:sunrpc                    *:*
LISTEN
tcp        0      0 *:58642                     *:*
LISTEN
tcp        0      0 *:ssh                       *:*
LISTEN
udp        0      0 *:35036                     *:*
udp        0      0 *:npmp-local                *:*

Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     16972  /tmp/orbit-
root/linc-76b-0-6fa08790553d6
unix  2      [ ACC ]     STREAM     LISTENING     17149  /tmp/orbit-
root/linc-794-0-7058d584166d2
unix  2      [ ACC ]     STREAM     LISTENING     17161  /tmp/orbit-
root/linc-792-0-546fe905321cc
unix  2      [ ACC ]     STREAM     LISTENING     15938  /tmp/orbit-
root/linc-74b-0-415135cb6a
```

Route :

About route

Show or manipulate the IP routing table.

Overview

In computer networking, a router is a device responsible for forwarding network traffic. When datagrams arrive at a router, the router must determine the best way to *route* them to their destination.

On Linux, BSD, and other Unix-like systems, the **route** command is used to view and make changes to the kernel routing table. The command syntax is different on different systems; here, when it comes to specific command syntax, we'll be discussing the Linux version.

Running **route** at the command line without any options will display the routing table entries:

```
route
```

```
Kernel IP routing table

Destination     Gateway         Genmask         Flags Metric Ref    Use Iface

default         192.168.1.2     0.0.0.0         UG    1024   0        0 eth0

192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
```

This shows us how the system is currently configured. If a packet comes into the system and has a destination in the range **192.168.1.0** through **192.168.1.255**, then it is forwarded to the gateway *****, which is **0.0.0.0** — a special address which represents an invalid or non-existant destination. So, in this case, our system will not route these packets.

route syntax

route [-CFvnee]

route [-v] [-A *family*] add [-net|-host] target [netmask *Nm*] [gw *Gw*]

[metric *N*] i [mss *M*] [window *W*] [irtt *m*] [reject] [mod] [dyn]

[reinstate] [[dev] *If*]

route [-v] [-A *family*] del [-net|-host] target [gw *Gw*] [netmask *Nm*]

[metric *N*] [[dev] *If*]

route [-V] [--version] [-h] [--help]

Technical Description

route manipulates the kernel's IP routing tables. Its primary use is to set up staticroutes to specific hosts or networks via an interface after it has been configured with the ifconfig program.

When the **add** or **del** options are used, **route** modifies the routing tables. Without these options, **route** displays the current contents of the routing tables.

Traceroute

The traceroute command is used in Linux to map the journey that a packet of information undertakes from its source to its destination. One use for traceroute is to locate when data loss occurs throughout a network, which could signify a nodethat's down.

Because each hop in the record reflects a new server or router between the originating PC and the intended target, reviewing the results of a traceroute scan also lets you identify slow points that may adversely affect your network traffic.

How It Works

Evaluating the specific route that network traffic follows (or finding the miscreant gateway that's discarding your packets) presents several troubleshooting challenges. Traceroute uses the IP protocol *time to live* field to solicit an ICMP TIME_EXCEEDED response from each gateway along the path to a destination host.

The only parameter you must include when you execute the traceroute command is the host name or IP address of the destination.

Traceroute Syntax and Switches

Traceroute Syntax in Ubuntu.

traceroute [**-dFInrvx**] [**-f** *first_ttl*] [**-g** *gateway*] [**-i** *iface*] [**-m** max_ttl] [**-p** *port*] [**-q** *nqueries*] [**-s** *src_addr*] [**-t** *tos*] [**-w** *waittime*] [**-z** *pausemsecs*] *host* [*packetlen*]

While the above is how the traceroute command has to be written out in order to work in the command line, the performance or output of the command can be changed by specifying one or more optional switches.

- **-f:** Set the initial time-to-live used in the first outgoing probe packet.
- **-F:** Set the "don't fragment" bit.
- **-d:** Enable socket level debugging.
- **-g:** Specify a loose source route gateway (8 maximum).
- **-i:** Specify a network interface to obtain the source IP address for outgoing probe packets. This is normally only useful on a multi-homed host. (See the **-s** flag for another way to do this.)
- **-I:** Use ICMP ECHO instead of UDP datagrams.
- **-m:** Set the max time-to-live (max number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).
- **-n:** Print hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).
- **-p:** Set the base UDP port number used in probes (default is 33434). Traceroute hopes that nothing is listening on UDP ports *base* to *base* + *nhops* - *1* at the destination host (so an ICMP PORT_UNREACHABLE message will be returned to terminate the

route tracing). If something is listening on a port in the default range, this option can be used to pick an unused port range.

- **-r:** Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (e.g., after the interface was dropped by *routed*(8C)).
- **-s:** Use the following IP address (which usually is given as an IP number, not a hostname) as the source address in outgoing probe packets. On multi-homed hosts (those with more than one IP address), this option can be used to force the source address to be something other than the IP address of the interface the probe packet is sent on. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent. (See the **-i** flag for another way to do this.)
- **-t:** Set the *type-of-service* in probe packets to the following value (default zero). The value must be a decimal integer in the range 0 to 255. This option can be used to see if different types-of-service result in different paths. (If you are not running 4.4bsd, this may be academic, since the normal network services like telnet and ftp don't let you control the TOS.) Not all values of TOS are legal or meaningful—see the IP spec for definitions. Useful values are probably `**-t** *16*' (low delay) and `**-t** *8*' (high throughput).
- **-v:** Verbose output. Received ICMP packets other than TIME_EXCEEDED and UNREACHABLEs are listed.
- **-w:** Set the time (in seconds) to wait for a response to a probe (default 5 sec.).
- **-x:** Toggle IP checksums. Normally, this prevents traceroute from calculating IP checksums. In some cases, the operating system can overwrite parts of the outgoing packet but not recalculate the checksum; thus, in some cases the default is to not calculate checksums and using **-x** causes them to be calculated. Note that checksums are usually required for the last hop when using ICMP ECHO probes (**-I**), so they are always calculated when using ICMP.
- **-z:** Set the time (in milliseconds) to pause between probes (default 0). Some systems such as Solaris and routers from Cisco, rate limit icmp messages. A good value to use with this is 500 (e.g., 1/2 second).

NSLOOKUP

The **nslookup** command is used to query Internet name servers interactively for information.

Overview

nslookup, which stands for "name server lookup", is a useful tool for finding out information about a named domain.

By default, **nslookup** will translate a domain name to an IP address (or vice versa). For instance, to find out what the IP address of **microsoft.com** is, you could run the command:

nslookup microsoft.com

...and you would receive a response like this:

Server: 8.8.8.8

Address: 8.8.8.8#53

Non-authoritative answer:

Name: microsoft.com

Address: 134.170.185.46

Name: microsoft.com

Address: 134.170.188.221

Here, **8.8.8.8** is the address of our system's Domain Name Server. This is the server our system is configured to use to translate domain names into IP addresses. "**#53**" indicates that we are communicating with it on port 53, which is the standard port number domain name servers use to accept queries.

Below this, we have our lookup information for **microsoft.com**. Our name server returned two entries, **134.170.185.46** and **134.170.188.221**. This indicates that **microsoft.com** uses a round robin setup to distribute server load. When you access **micrsoft.com**, you may be directed to either of these servers and your packets will be routed to the correct destination.

ARP

arp manipulates or displays the kernel's IPv4network neighbour cache. It can add entries to the table, delete one, or display the current content.

ARP stands for **Address Resolution Protocol**, which is used to find the address of a network neighbor for a given IPv4 address.

arp syntax

arp [-vn] [-H *type*] [-i *if*] -a [*hostname*]

arp [-v] [-i *if*] -d *hostname* [*pub*]

arp [-v] [-H *type*] [-i *if*] -s *hostname hw_addr* [temp]

arp [-v] [-H *type*] [-i *if*] -s *hostname hw_addr* [netmask *nm*] pub

arp [-v] [-H *type*] [-i *if*] -Ds *hostname ifa* [netmask *nm*] pub

arp [-vnD] [-H *type*] [-i *if*] -f [*filename*]

Modes

arp with no mode specifies will print the current content of the table. It is possible to limit the number of entries printed, by specifying an hardware address type, interface name or host address.

arp -d *address* will delete an ARP table entry. Root privilege is required to do this. The entry is found by IP address. If a hostname is given, it will be resolved before looking up the entry in the ARP table.

The command:

arp -s *address hw_addr*

is used to set up a new table entry. The format of the *hw_addr* parameter is dependent on the hardware class, but for most classes one can assume that the usual presentation can be used. For the Ethernet class, this is 6 bytes in hexadecimal, separated by colons. When adding proxy **arp** entries (that is those with the publish ("**pub**") flag set a **netmask** may be specified to proxy **arp** for entire subnets. This is not good practice, but is supported by older kernels because it can be useful. If the **temp** flag is not supplied entries will be permanent stored into the ARP cache. To simplify setting up entries for one of your network interfaces, you can use the "**arp -Ds** *address ifname*" form. In that case the hardware address is taken from the interface with the specified name.

FINGER

About finger

finger looks up and displays information about system users.

finger syntax

```
finger [-lmsp] [user ...] [user@host ...]
```

Options

-s	Displays the user's login name, real name, terminal name and write status (as a "*" after the terminal name if write login time, office location and office Login time is displayed as month, day, hours and minutes, unless more than six months ago, in which case the year i and

	Unknown devices as well as nonexistent idle and login times are displayed as single asterisks.
-l	Produces a multi-line format displaying all of the information described for the **-s** option as well as the user's home login shell, mail status, and the contents of the files ".plan", ".project", ".pgpkey" and ".forward" fr Phone numbers specified as eleven digits are printed as "**+N-NNN-NNN-NNNN**". Numbers specified as ten or appropriate subset of that string. Numbers specified as five digits are printed as "**xN-NNNN**". Numbers specifi "**xNNNN**". If write permission is denied to the device, the phrase "**(messages off)**" is appended to the line containing the dev displayed with the **-l** option; if a user is logged on multiple times, terminal information is Mail status is shown as "**No Mail.**" if there is no mail at all, "**Mail last read DDD MMM ## HH:MM YYYY (** their mailbox since new mail arriving, or "New mail received ...", " Unread since ..." if they have new mail.
-p	Prevents the **-l** option of **finger** from displaying the contents of the "**.plan**", "**.project**" and "**.pgpkey**" files.
-m	Prevent matching of usernames. The *user* is usually a login name; however, matching will also be done on the **m** option is supplied. All name matching performed by finger is case insensitive.

If no options are specified, **finger** defaults to the **-l** style output if operands are provided, otherwise to the **-s** style. Note that some fields may be missing, in either format, if information is not available for them.

If no arguments are specified, **finger** will print an entry for each user currently logged into the system.

Finger may be used to look up users on a remote machine. The format is to specify a user as "**user@host**", or "**@host**", where the default output format for the former is the **-l** style, and the default output format for the latter is the **-s** style. The **-l** option is the only option that may be passed to a remote machine.

If standard output is a socket, **finger** will emit a carriage return (^**M**) before every linefeed (^**J**). This format is for processing remote finger requests when invoked by **fingerd**, the finger daemon.

PORT SCAN / NMAP : Scanning network for open ports with nmap command

You can use nmap tool for this job. It is flexible in specifying targets. User can scan entire network or selected host or single server. Nmap is also useful to test your firewall rules. namp is metwork exploration tool and security / port scanner. According to nmap man page: It is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

map port scanning

TCP Connect scanning for localhost and network 192.168.0.0/24

```
# nmap -v -sT localhost
# nmap -v -sT 192.168.0.0/24
```

nmap TCP SYN (half-open) scanning

```
# nmap -v -sS localhost
# nmap -v -sS 192.168.0.0/24
```

nmap TCP FIN scanning

```
# nmap -v -sF localhost
# nmap -v -sF 192.168.0.0/24
```

nmap TCP Xmas tree scanning

Useful to see if firewall protecting against this kind of attack or not:

```
# nmap -v -sX localhost
# nmap -v -sX 192.168.0.0/24
```

nmap TCP Null scanning

Useful to see if firewall protecting against this kind attack or not:

```
# nmap -v -sN localhost
# nmap -v -sN 192.168.0.0/24
```

Conclusion : In this assignment we learnt different Linux commands used in network and their utilization with their options.