Tools of the Trade

Attention: The description of the commands inhere is not complete. It is given as a guideline and may be used as quick reference. To obtain more information about the capabilities and the syntax of these commands take a look at the `man` pages or simply use wikipedia or google.

Remark: Certain tools can be found in following directories: `/usr/sbin`, `/usr/local/bin`.

ping

The `ping` utility is used to check connectivity to a host (to check if a host operating and network connections are intact). A small packet is sent through the network to a particular IP address. This packet contains 64 bytes - 56 data bytes and 8 bytes of protocol header information. The host that sent the packet then waits (or 'listens') for a return packet. If the connections are good and the target host is up, a good return packet will be received. The `ping` can also measure the round trip time of a packet. We use the following forms of the command:

```
$ ping host
```

It gives only the answer if the connectivity to the host is OK or not.

```
$ ping -s packetsize host
```

Option `-s packetsize` specifies the number of data bytes to be sent. The default is 56.

```
$ ping -I interval host
```

Option `-i interval` specifies the interval between sending ping requests. The default is one second.

traceroute

The `traceroute` utility traces the route that an IP packet follows from your station to another internet host. It shows how a site is physically connected to the Internet. Along the way it gives an understanding of how networks inter-connect. This network utility can also measure the round trip time between your station and the intermediate routers along the path.

The traceroute command is used to discover the routes that packets actually take when traveling to their destination. The device (for example, a router or a PC) sends out a sequence of User Datagram Protocol (UDP) datagrams to an invalid port address at the remote host. The default IP probe packet which encapsulates the UDP datagram is 40 bytes.

Three datagrams are sent, each with a Time-To-Live (TTL) field value set to one. The TTL value of 1 causes the datagram to “timeout” as soon as it hits the first router in the path; this router then responds with an ICMP Time Exceeded Message (TEM) indicating that the datagram has expired.

Another three UDP messages are now sent, each with the TTL value set to 2, which causes the second router to return ICMP TEMs. This process continues until the packets actually reach the other destination. Since these datagrams are trying to access an invalid port at the destination host, ICMP Port Unreachable Messages are returned, indicating an unreachable port; this event signals the Traceroute program that it is finished.

The purpose behind this is to record the source of each ICMP Time Exceeded Message to provide a trace of the path the packet took to reach the destination.

```
$ traceroute host
```

telnet *(obsolete)*

The `telnet` utility is used to connect from one host to another (remote login) via the internet network. This command allows you to log onto machines around the world that you have accounts on or that allow public access. The most common way to use the command is:

```
$ telnet host [port]
```

`port` indicates a port number (address of an application); if it is not specified, then the default telnet port is used. After issuing the `telnet` command, you will be presented with a login prompt for the host system. You may now login to the host system. When you are finished with your remote session, logout as usual. The other way to use the command is to use the escape character Ctrl-] (control hook) to control telnet (make Ctrl-] and then ? to see the telnet commands).

ssh

SSH is a set of standards and an associated network protocol that allows establishing a secure channel between a local and a remote computer. The SSH client - ssh - that supports terminal protocols is typically used for remote administration of the SSH server computer via terminal console, thus it is used as an alternative to telnet. The most common way to use the command is:

```
$ ssh -l username hostname
```

or

```
$ ssh username@hostname
```

After issuing the ssh command, if the remote computer authenticates you successfully, you will be presented with a login prompt for the remote machine. You may now login to the host system. When you are finished with your remote session, logout as usual.

ifconfig

The **ifconfig** utility is used to assign an address to a network interface and to configure or display the current network interface configuration information. It must be used at system startup to define the network address of each interface present on a machine. After system startup, it can also be used to redefine an interface's address and its other operating parameters. 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. Otherwise, it configures an interface.

netstat

The netstat utility allows to print the various data related to the network configuration of a station. Here, we show only two forms of this command:

Option -i allows to print the state of the network interfaces (e.g. eth0 or lo).

```
$ netstat -ni
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500 0    633944      0      0      0   554412      0      0      0 BMRU
lo    16436 0     35118      0      0      0    35118      0      0      0 LRU
```

The MTU and Met fields show the current MTU and metric values for that interface. The RX and TX columns show how many packets have been received or transmitted error-free (RX-OK/TX-OK) or damaged (RX-ERR/TX-ERR); how many were dropped (RX-DRP/TX-DRP); and how many were lost because of an overrun (RX-OVR/TX-OVR).

The last column shows the flags that have been set for this interface. These characters are one-character versions of the long flag names that are printed when you display the interface configuration with **ifconfig**:

- B - A broadcast address has been set
- L - This interface is a loopback device
- M - All packets are received (promiscuous mode)
- O - ARP is turned off for this interface
- P - This is a point-to-point connection
- R - Interface is running
- U - Interface is up

When you invoke netsta with the -r flag, it displays the kernel routing table. The -n option makes netstat print addresses as dotted quad IP numbers rather than the symbolic host and network names. This option is especially useful when you want to avoid address lookups over the network (e.g., to a DNS or NIS server).

```
$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
128.178.156.0   0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         128.178.156.1   0.0.0.0         UG        0 0          0 eth0
```

The second column of netstat's output shows the gateway to which the routing entry points. If no gateway is used, an asterisk is printed instead. The third column shows the "generality" of the route, i.e., the network mask for this route. The fourth column displays the following flags that describe the route:

- G - The route uses a gateway
- U - The interface to be used is up
- H - Only a single host can be reached through the route
- D - This route is dynamically created. It is set if the table entry has been generated by a routing daemon like gated or by an ICMP redirect message
- M - This route is set if the table entry was modified by an ICMP redirect message
- ! - The route is a reject route and datagrams will be dropped

netperf

The netperf utility provides tests for both unidirectional throughput and end-to-end latency. The calculation of these parameters is based on measuring the time necessary to transmit messages of fixed size by protocol TCP.

The most common use of `netperf` is measuring bulk data transfer performance. This is also referred to as “stream” or “unidirectional stream” performance. Essentially, these tests will measure how fast one system can send data to another and/or how fast that other system can receive it. Request/response performance is the second area that can be investigated with `netperf`. Generally speaking, `netperf` request/response performance is quoted as “transactions/s” for a given request and response size. A transaction is defined as the exchange of a single request and a single response. From a transaction rate, one can infer one way and round-trip average latency.

```
$ netperf [host [message_size [count]]]
```

The obligatorily parameter is the name or address of a station destination towards which we want to measure the performance. The optional parameters are the size of the messages (in Bytes, by default 1024) and the number of messages to be sent (if the latter is not specified, netperf sends messages until you stop it by Ctrl-C).

The `netperf` prints a statistics report regularly (approximately every second) in a table of 5 columns with the following heading:

```
Total Delta    |    Avg 1    Avg 10    Avg (kbits/s)
```

“Total” is the total number of messages sent since the launching of netperf; “Delta” is the number of messages sent since the preceding report/ratio; “Avg 1” is the average throughput since the preceding report; “Avg 10” is the average throughput during the last 10 reports; “Avg” is the average throughput since the launching of netperf. All the throughputs are expressed in Kbit/s.

nslookup, dig, host

The `nslookup` utility is used to find out the corresponding IP address of a host name (e.g., “whatis.com”) by contacting Internet Domain Name Servers. It also does reverse name lookup and find the host name for an specified IP address. It has two modes: interactive and non-interactive. Interactive mode is used to get information about various hosts and domains or to displays a list of hosts in a domain. Non-interactive mode is used to display just the name and requested information for a host or domain.

Noninteractive mode:

```
$ nslookup host
```

Interactive mode:

```
$ nslookup
> server ns1.nic.fr
> set q=NS
> imag.fr.
Server:  ns1.nic.fr
Address:  192.93.0.1

Non-authoritative answer:
imag.fr nameserver = dns.inria.fr
imag.fr nameserver = imag.imag.fr
imag.fr nameserver = isis.imag.fr
imag.fr nameserver = ns2.nic.fr

Authoritative answers can be found from:
dns.inria.fr    internet address = 193.51.208.13
imag.imag.fr    internet address = 129.88.30.1
isis.imag.fr    internet address = 129.88.32.24
ns2.nic.fr      internet address = 192.93.0.4
```

The `dig` command is similar to `nslookup`. It is a tool for querying DNS nameservers for information about host addresses, mail exchanges, nameservers, and related information. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. As with `nslookup`, `dig` provides the possibiliy of inverse DNS queries through when the `-x` option is given.

Most DNS administrators use `dig` to troubleshoot DNS problems because of its flexibility, ease of use and clarity of output. Other lookup tools tend to have less functionality than `dig`. Although `dig` is normally used with command-line arguments, it also has a batch mode of operation for reading lookup requests from a file. A brief summary of its command-line arguments and options is printed when the `-h` option is given. A further description may be found at the `man` page or here.

Below, you may also find a short description of the sections which are returned by the `dig` command.

- HEADER: This displays the dig command version number, the global options used by the dig command, and few additional header information.
- QUESTION SECTION: This displays the question it asked the DNS. i.e This is one's input. If one has given the command `dig WEBSITE`, `dig` will indicate in this section the WEBSITE (e.g. cnn.com) that we asked for along with the type of the query. The default type `dig` command uses is "A record".
- ANSWER SECTION: This displays the answer it receives from the DNS. i.e This is the output. For the default query type, this displays the "A record" of the WEBSITE (e.g. cnn.com).

- AUTHORITY SECTION: This displays the DNS name server that has the authority to respond to this query. Basically this displays available name servers of the website.
- ADDITIONAL SECTION: This displays the ip address of the name servers listed in the AUTHORITY SECTION. Stats section at the bottom displays few dig command statistics including how much time it took to execute this query.

Another useful command, but with lower functionality of both `nslookup` and `dig` is the `host` command. It is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa. When no arguments or options are given, host prints a short summary of its command line arguments and options.

grep

The `grep` utility searches text files for a pattern and prints all lines that contain that pattern. If you want to search for a fixed string you should rather use `fgrep`. For example, to find which port protocol SMTP uses type the following:

```
$ fgrep smtp /etc/services
smtp            25/tcp          mail
ssmtp           465/tcp                         # SMTP over SSL
```

Wireshark (*formerly known as* Ethereal)

Wireshark is a free software protocol analyzer, or packet sniffer application, used for network troubleshooting, analysis, software and protocol development, and education. It has all of the standard features of a protocol analyzer. The functionality it provides is very similar to `tcpdump`, but it has a GUI front-end, and many more information sorting and filtering options. It allows the user to see all traffic being passed over the network by putting the network card into promiscuous mode. Moreover, it “understands” the structure of different network protocols and thus it’s able to display encapsulation and single fields and interpret their meaning. Here you can find the official Wireshark User’s Guide