Cheat Sheet

Networking commands in UNIX

DESCRIPTION

All of the commands which take filename parameters may use wildcards in those positions, but remember to use the shell's single and double quotes to prevent wildcard expansion when parameters contain spaces, tabs, etc.

`ifconfig`

Display status of all configured network interfaces.

`ifconfig` *interface*

Display the status of the interface provided; if the interface is not configured, the `flags` will not include the word **`UP`**.

`ifconfig` *interface ipaddr* `netmask` *mask*

Configures the given *interface* to the specified IP address (*ipaddr*) and assigns *mask* as the subnet mask. A route for the network (through *interface*) will be automatically added to the routing table.

`ifconfig` *interface:0 ipaddr* `netmask` *mask*

Configures an alias (in the example, alias `:0`) for the given *interface* to the specified IP address (*ipaddr*) and assigns *mask* as the subnet mask. Aliases can range from `:0` to `:255`. A route for the network (through *interface*) will be automatically added to the routing table.

`ifconfig` *interface* `down`

Bring down an interface without changing any of its configuration parameters. This allows re-enabling the interface by running the command again, substituting `up` for `down`.

`route [-n]`

Displays the contents of the routing table, attempting to perform reverse-DNS lookups so that IP addresses can be displayed as hostnames. To prevent the reverse lookup, add the `-n` option.

`ping` *hostname*

Tests the network connectivity from the current machine to *hostname*. Hostname must be resolvable to an IP address using DNS, `/etc/hosts`, or some other technique. (Or the provided value can be an IP address directly.)

Note that the `ping` command is implemented using **ICMP**, so all of the proper route statements must be in place between the two endpoints. (In other words, the ping packet may reach the destination but not be able to return due to incorrect route table entries somewhere along the packet's path.)

`traceroute` *hostname*

Similar to `ping`, except that this command sets the lifetime of the packet to just one hop and prints out the IP address of the host that returns the **ICMP** error packet. It then increments the lifetime by one and attempts delivery again, printing the IP address of the host that returns the error packet. This continues until the error packet comes from the destination host itself.

In some environments, the `ping -R` command may provide similar functionality.

`netstat [-a] [-n]`

This command displays the status of current network connections. By default, only established TCP connections and local UNIX domain sockets are displayed. Adding the `-a` option causes it to display all sockets, including those merely waiting for an incoming connection (such as all servers). Adding `-n` prevents the reverse-DNS lookup that tries to convert IP addresses to hostnames (thus displaying the numeric IP addresses themselves).

`nmap`

`Nmap` is used to scan one or more machines by subnet address or domain. Running this command on an organization's network will likely be treated as "computer trespass" as defined by United States federal statute as a criminal, and possibly terrorist, action. (Pretty sad, eh?)

See the usage message from `nmap` for details on options.

`strace -p` *pid*

Displays a *system call* trace for the specified process ID. System calls are the implementation of the POSIX API (this is a generalization, but a useful one). After attaching to the given process, the `strace` command will not terminate until the traced process does. Pressing **Ctrl**-**C** will terminate `strace` without terminating the traced process.

`strace` *command*

Same as the above, but this syntax is used when the program to be traced is not yet running. Pressing **Ctrl-C** in this case will send the **SIGINT** signal to the *command* itself, which will terminate most applications (and hence, the `strace` will terminate as well).

`ltrace -p` *pid*

Similar to `strace`, except that it traces *library calls* instead. Also note that the `-s` option to `ltrace` causes it to display output similar to `strace`, yet that output will be intermingled with the normal library call output, allowing the user to see which system calls are invoked by the library calls.

`ltrace` *command*

Same as `ltrace`, but traces the given command.

`tcpdump -i` *interface filter*

Displays a packet dump to stdout in a somewhat complex text form. The *interface* is the same as used in the `ifconfig` command. The *filter* is a sophisticated expression that limits the packet logging to only particular packets, thus reducing the output of the program and the CPU load on the machine.

The packet logged can also be written to a file (using the `-w` option) to be processed later, possibly on a different machine or using a graphical interface (such as `ethereal`, next).

`ethereal`

This is a graphical version of `tcpdump` with the same filtering capabilities.

`iptables`

Configure the kernel-based firewall filter rules. This is an extremely complex tool and full documentation can be found as `iptables`(8) in the man pages.