Foreword

This guide stems from the notes I have been taking while studying and working with Linux. It contains useful information about standards and tools for Linux system administration, as well as a good amount of topics from the certification exams LPIC-1 (Linux Professional Institute Certification level 1), LPIC-2, RHCSA (Red Hat Certified System Administrator), RHCE (Red Hat Certified Engineer), and CEH (Certified Ethical Hacker). Unless otherwise specified, the shell of reference is Bash.

This is an independent publication and is not affiliated with LPI, Red Hat, EC-Council, or any other organization. You can freely print, use, and share the whole guide or the single pages, provided that you distribute them unmodified and not for profit.

This document has been composed with Apache OpenOffice.

Happy Linux hacking,

Daniele Raffo

Version history

1st edition May 2013
2nd edition September 2014
3rd edition July 2015
4th edition June 2016
5th edition September 2017
6th edition August 2018
7th edition May 2019
8th edition January 2020

9th edition January 2021

Bibliography and suggested readings

- Evi Nemeth et al., *UNIX and Linux System Administration Handbook*, O'Reilly
- Adam Haeder et al., *LPI Linux Certification in a Nutshell*, O'Reilly
- Christoph Braun, *Unix System Security Essentials*, Addison-Wesley
- The Linux Documentation Project guides, [https://www.tldp.org/guides.html](https://www.tldp.org/guides.html)
- A-Z index of Bash command line, [http://ss64.com/bash](http://ss64.com/bash)
- Shell command line snippets, [http://www.commandlinefu.com](http://www.commandlinefu.com)
- Bash command line snippets, [http://www.bashoneliners.com](http://www.bashoneliners.com)
- RAM management in Linux, [http://www.linuxatemyram.com](http://www.linuxatemyram.com)
- Regular expressions tester and cheat sheet, [http://www.regextester.com](http://www.regextester.com)
- Bash pitfalls, [http://mywiki.wooledge.org/BashPitfalls](http://mywiki.wooledge.org/BashPitfalls)
- CentOS 7 man pages, [https://www.unix.com/man-page-centos-repository.php](https://www.unix.com/man-page-centos-repository.php)
- Ansible documentation, [https://docs.ansible.com/ansible/latest](https://docs.ansible.com/ansible/latest)
Logical Volume Management (LVM) introduces an abstraction between physical and logical storage, allowing a more versatile use of filesystems. LVM uses the Linux device mapper feature (/dev/mapper).

Disks, partitions, and RAID devices are made of **Physical Volumes**, which are grouped into a **Volume Group**. A Volume Group is divided into small fixed-size chunks called Physical Extents, which are mapped 1-to-1 to Logical Extents. Logical Extents are grouped into **Logical Volumes**, on which filesystems are created.

### How to create a Logical Volume

1. Add a new disk to the machine
2. `lsblk`  
   Verify that the new disk is recognized e.g. as `/dev/sda`
3. `fdisk /dev/sda`  
   Create a new partition (of type 0x8E = Linux LVM) on the new disk. This is not necessary but recommended, because other OSes might not recognize the LVM header and see the whole unpartitioned disk as empty
4. `pvcreate /dev/sda1`  
   Initialize the Physical Volume to be used with LVM
5. `vgcreate -s 8M myvg0 /dev/sda1`  
   Create a Volume Group and define the size of Physical Extents to 8 Mb (default value is 4 Mb)
   
   or
   
   `vgextend myvg0 /dev/sda1`  
   or add the Physical Volume to an existing Volume Group
6. `mkfs -t ext3 /dev/myvg0/mylv`  
   Create a filesystem on the Logical Volume
7. `mount /dev/myvg0/mylv /mnt/mystuff`  
   Mount the Logical Volume

### How to increase the size of a Logical Volume (operation possible only if the underlying filesystem allows it)

1. Add a new disk to the machine, to provide the extra disk space
2. `pvcreate /dev/sdc`  
   Initialize the Physical Volume
3. `vgextend myvg0 /dev/sdc`  
   Add the Physical Volume to an existing Volume Group
   
   or
   
   1. Increase the size of an existing disk (already initialized as PV)
   2. `partprobe`  
      Notify the kernel of the new disk size
   3. `pvresize /dev/sdc`  
      Accommodate the Physical Volume to the new size

   Then:
   
   4. `lvextend -L 2048M /dev/myvg0/mylv`  
      Extend the Logical Volume by 2 Gb
   
   or
   
   3. `lvresize -l+100%FREE /dev/myvg0/mylv`  
      or extend the Logical Volume taking all free space
   5. `resize2fs /dev/myvg0/mylv (ext)`  
      Extend the filesystem.
   
   xfs_growfs /dev/myvg0/mylv (XFS)  
   Alternatively, use `lvresize -r` on the previous step

### How to reduce the size of a Logical Volume (operation possible only if the underlying filesystem allows it)

1. `resize2fs /dev/myvg0/mylv 900M`  
   Shrink the filesystem to 900 Mb
2. `lvreduce -L 900M /dev/myvg0/mylv`  
   Shrink the Logical Volume to 900 Mb

### How to snapshot and backup a Logical Volume

1. `lvcreate -s -L 1024M -n mysnap /dev/myvg0/mylv`  
   Create the snapshot like a Logical Volume
2. `tar cvzf mysnap.tar.gz mysnap`  
   Backup the snapshot with any backup tool
3. `lvremove /dev/mvvg0/mysnap`  
   Delete the snapshot
### LVM - commands

<table>
<thead>
<tr>
<th>PV commands</th>
<th>VG commands</th>
<th>LV commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvs</td>
<td>vgs</td>
<td>lvs</td>
</tr>
<tr>
<td>pvscan</td>
<td>vgscan</td>
<td>lvscan</td>
</tr>
<tr>
<td>pvdisplay</td>
<td>vgdisplay</td>
<td>lvdisplay</td>
</tr>
<tr>
<td>pvck</td>
<td>vgck</td>
<td></td>
</tr>
<tr>
<td>pvcreate</td>
<td>vgcreate</td>
<td>lvcreate</td>
</tr>
<tr>
<td>pvchange</td>
<td>vgchange</td>
<td>lvchange</td>
</tr>
<tr>
<td>pvremove</td>
<td>vgremove</td>
<td>lvremove</td>
</tr>
<tr>
<td>pvresize</td>
<td>vgmerge</td>
<td>lvresize</td>
</tr>
<tr>
<td>pvmove</td>
<td>vgsplit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vgsplit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vgimport</td>
<td></td>
</tr>
<tr>
<td></td>
<td>vgexport</td>
<td></td>
</tr>
<tr>
<td>/dev/mapper/vgname-lvname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/vgname/lvname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/etc/lvm/archive/</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### PV commands
- **pvs**: Report information about Physical Volumes
- **pvscan**: Scan all disks for Physical Volumes
- **pvdisplay**: Display Physical Volume attributes
- **pvck**: Check Physical Volume metadata
- **pvcreate**: Initialize a disk or partition for use with LVM
- **pvchange**: Change Physical Volume attributes
- **pvremove**: Remove a Physical Volume
- **pvresize**: Modify the size of a Physical Volume
- **pvmove**: Move the Logical Extents on a Physical Volume to wherever there are available Physical Extents (within the Volume Group) and then put the Physical Volume offline

### VG commands
- **vgs**: Report information about Volume Groups
- **vgscan**: Scan all disks for Volume Groups
- **vgdisplay**: Display Volume Group attributes
- **vgck**: Check Volume Group metadata
- **vgcreate**: Create a Volume Group using Physical Volumes
- **vgchange**: Change Volume Group attributes
- **vgremove**: Remove a Volume Group
- **vgextend**: Add a Physical Volume to a Volume Group
- **vgreduce**: Remove a Physical Volume from a Volume Group
- **vgmerge**: Merge two Volume Groups
- **vgsplit**: Split two Volume Groups
- **vgimport**: Import a Volume Group into a system
- **vgexport**: Export a Volume Group from a system

### LV commands
- **lvs**: Report information about Logical Volumes
- **lvscan**: Scan all disks for Logical Volumes
- **lvdisplay**: Display Logical Volume attributes
- **lvcreate**: Create a Logical Volume in a Volume Group
- **lvchange**: Change Logical Volume attributes
- **lvremove**: Remove a Logical Volume
- **lvextend**: Increase the size of a Logical Volume
- **lvreduce**: Shrink the size of a Logical Volume
- **lvresize**: Modify the size of a Logical Volume

### LVM global commands
- **dmsetup command**: Perform low-level LVM operations
- **lvm command**: Perform LVM operations. May also be used as an interactive tool
- **lvmsar**: LVM system activity reporter. Unsupported on LVM2
- **lvmdiskscan**: Scan the system for disks and partitions usable by LVM
- **lvmconfig**: Show the current LVM disk configuration

/`/dev/mapper/vgname-lvname` Mapping of Logical Volumes in the filesystem
/`/dev/vgname/lvname` Directory containing Volume Groups metadata backups
<table>
<thead>
<tr>
<th>Boot sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POST</strong> (Power-On Self Test)</td>
</tr>
<tr>
<td><strong>BIOS</strong> (Basic I/O System)</td>
</tr>
<tr>
<td><strong>Chain loader GRUB</strong> (GRand Unified Bootloader)</td>
</tr>
<tr>
<td><strong>Linux kernel</strong></td>
</tr>
<tr>
<td><strong>init</strong></td>
</tr>
<tr>
<td><strong>Startup</strong></td>
</tr>
<tr>
<td><strong>Login</strong></td>
</tr>
</tbody>
</table>

Modern systems use UEFI (Unified Extensible Firmware Interface) instead of BIOS. UEFI does not use the MBR boot code; it has knowledge of partition table and filesystems, and stores its application files required for launch in an EFI System Partition, mostly formatted as FAT32. After the POST, the system loads the UEFI firmware which initializes the hardware required for booting, then reads its Boot Manager data to determine which UEFI application to launch. The launched UEFI application may then launch another application, e.g. the kernel and initramfs in case of a boot loader like GRUB.

Information about the boot process can be found in the manpages `man 7 boot` and `man 7 bootup`. 
SysV startup sequence

<table>
<thead>
<tr>
<th>Startup sequence</th>
<th>Debian</th>
<th>Red Hat</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>At startup</strong> /sbin/init executes all instructions on /etc/inittab. This script at first switches to the default runlevel...</td>
<td>id:2:initdefault:</td>
<td>id:5:initdefault:</td>
</tr>
<tr>
<td>... then it runs the following script (same for all runlevels) which configures peripheral hardware, applies kernel parameters, sets hostname, and provides disks initialization...</td>
<td>/etc/init.d/rcS</td>
<td>/etc/rc.d/rc.sysinit or /etc/rc.sysinit</td>
</tr>
<tr>
<td>... and then, for runlevel N, it calls the script /etc/init.d/rc N (i.e. with the runlevel number as parameter) which launches all services and daemons specified in the following startup directories:</td>
<td>/etc/rcN.d/</td>
<td>/etc/rc.d/rcN.d/</td>
</tr>
</tbody>
</table>

The startup directories contain symlinks to the init scripts in /etc/init.d/ which are executed in numerical order.

Links starting with K are called with argument stop, links starting with S are called with argument start.

```
lrwxrwxrwx. 1 root root 14 Feb 11 22:32 K88sssd -> ../init.d/sssd
lrwxrwxrwx. 1 root root 15 Nov 28 14:50 K89rdisc -> ../init.d/rdisc
lrwxrwxrwx. 1 root root 17 Nov 28 15:01 S01sysstat -> ../init.d/sysstat
lrwxrwxrwx. 1 root root 18 Nov 28 14:54 S05cgconfig -> ../init.d/cgconfig
lrwxrwxrwx. 1 root root 16 Nov 28 14:52 S07iscsid -> ../init.d/iscsid
lrwxrwxrwx. 1 root root 18 Nov 28 14:42 S08iptables -> ../init.d/iptables
```

The last script to be run is S99local -> ../init.d/rc.local; therefore, an easy way to run a specific program upon boot is to call it from this script file.

```
/proc/1/local
/proc/1/before.local (SUSE) runs only at boot time, not when switching runlevel.
/proc/1/after.local (SUSE) runs only at boot time, before the scripts in the startup directories.
```

To add or remove services at boot sequence:

```
update-rc.d service defaults
update-rc.d -f service remove
chkconfig --add service
chkconfig --del service
```

When adding or removing a service at boot, startup directories will be updated by creating or deleting symlinks for the default runlevels: K symlinks for runlevels 0 1 6, and S symlinks for runlevels 2 3 4 5.
Service will be run via the xinetd super server.

---

**Linux Standard Base (LSB)**

The Linux Standard Base defines a format to specify default values on an init script /etc/init.d/foo:

```
### BEGIN INIT INFO
# Provides: foo
# Required-Start: bar
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Description: Service Foo init script
### END INIT INFO
```

Default runlevels and S/K symlinks values can also be specified as such:

```
# chkconfig: 2345 85 15
# description: Foo service
```
Login

/etc/init/start-ttys.conf (Red Hat)
Start the specified number of terminals at bootup via getty, which manages physical or virtual terminals (TTYS)

/etc/sysconfig/init (Red Hat)
Control appearance and functioning of the system during bootup

/etc/machine-id (Red Hat)
Randomly-generated machine ID.
The machine ID can be safely regenerated by deleting this file and then running the command systemd-machine-id-setup

/etc/securetty
List of TTYS from which the root user is allowed to login

/etc/issue
Message printed before the login prompt. Can contain these escape codes:
\b Baudrate of line
\d Date
\s System name and OS
\l Terminal device line
\m Machine architecture identifier
\n Nodename aka hostname
\o Domain name
\r OS release number
\t Time
\u Number of users logged in
\U "n users" logged in
\v OS version and build date

/etc/issue.net
Message printed before the login prompt on a remote session

/etc/motd
Message Of The Day, printed after a successful login, but before execution of the login shell

/etc/nologin
If this file exists, login and sshd deny login to all unprivileged users.
Useful when doing system maintenance

/etc/login.defs
Definition of default values (UID and GID ranges, mail directory, account validity, password encryption method, etc.) for user account creation

/var/log/secure (Red Hat)
Logfile containing user logins (both successful and failed) and authentication mechanisms

/var/log/auth.log (Debian)
Logfile containing failed authentication attempts

To prevent a specific user to log in, their shell can be set either as:

/bin/false user is forced to exit immediately
/sbin/nologin user is prompted a message and forced to exit; the message is “This account is currently not available” or the contents of file /etc/nologin.txt if it exists

last
Print the list of users that logged in and out. Searches through the file /var/log/wtmp

lastb
Print the list of bad login attempts. Searches through the file /var/log/btmp

fail2ban
Temporarily ban IP addresses (via firewall rules) that have too many failed password logins. This information is taken from authentication logs

pam_tally2
Deny access to users that have too many failed logins

acct on
acct off
Turn process accounting on or off

ac
Print statistics about connect time of users

lastcomm
Print information about previously executed commands

sa
Print summarized information about previously executed commands
Runlevels

<table>
<thead>
<tr>
<th>Runlevel (SysV)</th>
<th>Target (Systemd)</th>
<th>Debian</th>
<th>Red Hat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Shutdown</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Single user / maintenance mode</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>multi-user.target</td>
<td>Multi-user mode (default)</td>
<td>Multi-user mode without network</td>
</tr>
<tr>
<td>3</td>
<td>multi-user.target</td>
<td>Multi-user mode</td>
<td>Multi-user mode with network</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Multi-user mode</td>
<td>Unused, for custom use</td>
</tr>
<tr>
<td>5</td>
<td>graphical.target</td>
<td>Multi-user mode</td>
<td>Multi-user mode with network and X (default)</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Reboot</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>Single user / maintenance mode (usually accessed through runlevel 1)</td>
<td></td>
</tr>
</tbody>
</table>

**Systemd's target** runlevel\(n\).\ target emulates a SysV's runlevel \(n\).

runlevel
who \(-r\)
init runlevel
telinit runlevel

systemctl get-default
systemctl set-default target
systemctl isolate target
systemctl emergency
systemctl rescue
systemctl \(-t\) target

init 0
telinit 0
shutdown \(-h\) now
halt
poweroff
systemctl isolate shutdown.target

init 6
telinit 6
shutdown \(-r\) now
reboot
systemctl isolate reboot.target

shutdown
down
shutdown \(-a\)
shutdown \(-h\) 16:00 \(message\)
shutdown \(-f\)
shutdown \(-F\)
shutdown \(-c\)

Shut down the system securely: all logged in users are notified via a message to their terminal, and login is disabled. Can only be run by the root user.

Non-root users that are listed in \(/etc/shutdown.allow\) can use this command to shut down the system.

Schedule a shutdown for 4 PM and send a warning message to all logged in users.

Skip fsck on reboot.

Force fsck on reboot.

Cancel a shutdown that has been already initiated.
SysV service management

/`etc/init.d/service operation`  
`service service operation`  
`rcservice operation`  

(Debian)  
Add a service at boot  

(Debian)  
Remove a service at boot  

Add a service on the default runlevels; creates S30 symlinks for starting the service and K70 symlinks for stopping it  

Add the service on runlevels 2 4 5  

Add the service on default runlevels  

Remove the service on default runlevels  

Check if the service is enabled on the current runlevel  

Reset the on/off state of the service for all runlevels to whatever the LSB specifies in the init script  

Reset the start/stop priorities of the service for all runlevels to whatever the LSB specifies in the init script  

Display current configuration of service (its status and the runlevels in which it is active)  

List all active services and their current configuration  

List services started on runlevel n

<table>
<thead>
<tr>
<th>Supported service operations</th>
<th>Mandatory/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Start the service</td>
</tr>
<tr>
<td>stop</td>
<td>Stop the service</td>
</tr>
<tr>
<td>restart</td>
<td>Restart the service (stop, then start)</td>
</tr>
<tr>
<td>status</td>
<td>Display daemon PID and execution status</td>
</tr>
<tr>
<td>force-reload</td>
<td>Reload configuration if service supports it, otherwise restart</td>
</tr>
<tr>
<td>condrestart</td>
<td></td>
</tr>
<tr>
<td>try-restart</td>
<td>Restart the service only if already running</td>
</tr>
<tr>
<td>reload</td>
<td>Reload the service configuration</td>
</tr>
</tbody>
</table>

|
**Systemd service management**

`systemctl operation service`  
Perform the specified operation (start, stop, status, etc.) on the specified service (unit file)

`systemctl enable service`  
Add the service on the current target

`systemctl disable service`  
Remove the service on the current target

`systemctl is-enabled service`  
Check if the service is enabled on the current target

`systemctl mask service`  
Mask the service on the current target. This prevents the service to be enabled or started

`systemctl unmask service`  
Unmask the service on the current target

`systemctl list-unit-files --type=service`  
List all active services and their current configuration

`systemctl`  
List loaded and active units

`systemctl --all`  
List all units, including inactive ones
/etc/inittab

# The default runlevel.
id:2:initdefault:

# Boot-time system configuration-initialization script.
# This is run first except when booting in emergency (-b) mode.
# Assignment of an init script to run at boot.
si::sysinit:/etc/init.d/rcS

# What to do in single-user mode.
~~:S:wait:/sbin/sulogin

# /etc/init.d executes the S and K scripts upon change of runlevel.
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
z6:6:respawn:/sbin/sulogin

# Normally not reached, but fall through in case of emergency.
# /sbin/getty invocations for the runlevels.
# Id field must be the same as the last characters of the device (after "tty").
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2

/etc/inittab describes which processes are started at bootup and during normal operation; it is read and executed by \texttt{init} at bootup. All its entries have the form \texttt{id:runlevels:action:process}.

<table>
<thead>
<tr>
<th>id</th>
<th>Runlevels for which the specified action must be performed. If empty, action is performed on all runlevels</th>
</tr>
</thead>
<tbody>
<tr>
<td>runlevels</td>
<td>respawn: Process will be restarted when it terminates. Process is started at the specified runlevel and init will wait for its termination (i.e. execution of further lines of /etc/inittab stops until the process exits)</td>
</tr>
<tr>
<td>action</td>
<td>wait: Process is executed once at the specified runlevel.</td>
</tr>
<tr>
<td></td>
<td>once: Process is executed at system boot. Runlevels field is ignored</td>
</tr>
<tr>
<td></td>
<td>boot: Process is executed at system boot and init will wait for its termination. Runlevels field is ignored</td>
</tr>
<tr>
<td></td>
<td>bootwait: Does nothing</td>
</tr>
<tr>
<td></td>
<td>off: Process is executed when an on-demand runlevel (A, B, C) is called</td>
</tr>
<tr>
<td></td>
<td>ondemand: Specifies the default runlevel to boot on. Process field is ignored</td>
</tr>
<tr>
<td></td>
<td>initdefault: Process is executed at system boot, before any boot or bootwait entries. Runlevels field is ignored</td>
</tr>
<tr>
<td></td>
<td>sysinit: Process is executed when power goes down and a UPS kicks in. init will not wait for its termination</td>
</tr>
<tr>
<td></td>
<td>powerfail: Process is executed when power goes down and a UPS kicks in. init will wait for its termination</td>
</tr>
<tr>
<td></td>
<td>powerwait: Process is executed when power is down and the UPS battery is almost empty</td>
</tr>
<tr>
<td></td>
<td>powerfailnow: Process is executed when power has been restored from UPS</td>
</tr>
<tr>
<td></td>
<td>powerokwait: Process is executed when init receives a SIGINT via CTRL + ALT + DEL</td>
</tr>
<tr>
<td></td>
<td>ctrlaltdel: Process is executed when a special key combination is pressed on console</td>
</tr>
<tr>
<td>process</td>
<td>Process to execute. If prepended by a +, utmp and wtmp accounting will not be performed</td>
</tr>
</tbody>
</table>
Filesystem Hierarchy Standard

<table>
<thead>
<tr>
<th>Directory</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>/bin</td>
<td>Essential command binaries for all users</td>
</tr>
<tr>
<td>/boot</td>
<td>Bootloader files (OS loader, kernel image, initrd, etc.)</td>
</tr>
<tr>
<td>/dev</td>
<td>Virtual filesystem containing device nodes to devices and partitions</td>
</tr>
<tr>
<td>/etc</td>
<td>System configuration files and scripts</td>
</tr>
<tr>
<td>/home</td>
<td>Home directories for users</td>
</tr>
<tr>
<td>/lib</td>
<td>Libraries for the binaries in /bin and /sbin, kernel modules</td>
</tr>
<tr>
<td>/lost+found</td>
<td>Storage directory for recovered files in this partition</td>
</tr>
<tr>
<td>/media</td>
<td>Mount points for removable media</td>
</tr>
<tr>
<td>/mnt</td>
<td>Mount points for temporary filesystems</td>
</tr>
<tr>
<td>/net</td>
<td>Access to directory tree on different external NFS servers</td>
</tr>
<tr>
<td>/opt</td>
<td>Optional, large add-on application software packages</td>
</tr>
<tr>
<td>/proc</td>
<td>Virtual filesystem providing kernel and processes information</td>
</tr>
<tr>
<td>/root</td>
<td>Home directory for the root user</td>
</tr>
<tr>
<td>/run</td>
<td>Runtime variable data; replaces /var/run</td>
</tr>
<tr>
<td>/sbin</td>
<td>Essential system binaries, system administration commands</td>
</tr>
<tr>
<td>/srv</td>
<td>Data for services provided by the system</td>
</tr>
<tr>
<td>/sys</td>
<td>Virtual filesystem providing information about hotplug hardware devices</td>
</tr>
<tr>
<td>/tmp</td>
<td>Temporary files; deleted at reboot</td>
</tr>
<tr>
<td>/usr</td>
<td>User utilities and applications</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/bin</td>
<td>Non-essential command binaries for all users</td>
</tr>
<tr>
<td>/usr/include</td>
<td>C header files</td>
</tr>
<tr>
<td>/usr/lib</td>
<td>Libraries for the binaries in /usr/bin and /usr/sbin</td>
</tr>
<tr>
<td>/usr/local</td>
<td>Software installed locally</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>/usr/local/bin</td>
<td>Local software binaries</td>
</tr>
<tr>
<td>/usr/local/games</td>
<td>Local game binaries</td>
</tr>
<tr>
<td>/usr/local/include</td>
<td>Local C header files</td>
</tr>
<tr>
<td>/usr/local/lib</td>
<td>Local libraries for the binaries in /usr/local/bin and /usr/local/sbin</td>
</tr>
<tr>
<td>/usr/local/man</td>
<td>Local man pages</td>
</tr>
<tr>
<td>/usr/local/sbin</td>
<td>Local system binaries</td>
</tr>
<tr>
<td>/usr/local/share</td>
<td>Local architecture-independent hierarchy</td>
</tr>
<tr>
<td>/usr/local/src</td>
<td>Local source code</td>
</tr>
<tr>
<td>/usr/sbin</td>
<td>Non-essential system binaries (daemons and services)</td>
</tr>
<tr>
<td>/usr/share</td>
<td>Architecture-independent files (icons, fonts, documentation, etc.)</td>
</tr>
<tr>
<td></td>
<td>Package-specific documentation not included in man pages</td>
</tr>
<tr>
<td>/usr/share/doc</td>
<td>Package-specific documentation not included in man pages</td>
</tr>
<tr>
<td>/usr/share/man</td>
<td>Man pages</td>
</tr>
<tr>
<td>/usr/share/info</td>
<td>Documentation in Info format</td>
</tr>
<tr>
<td>/usr/src</td>
<td>Source code for the current OS</td>
</tr>
<tr>
<td>/var</td>
<td>Variable files (logs, caches, mail spools, etc.)</td>
</tr>
<tr>
<td>/var/log</td>
<td>Logfiles</td>
</tr>
<tr>
<td>/var/opt</td>
<td>Variable files for the application software installed in /opt</td>
</tr>
<tr>
<td>/var/spool</td>
<td>Queued items to be processed (mail messages, cron jobs, print jobs, etc.)</td>
</tr>
<tr>
<td>/var/tmp</td>
<td>Temporary files that need to be stored for a longer time; preserved between reboots</td>
</tr>
</tbody>
</table>

The manpage `man hier` contains information about filesystem hierarchy.
The **superblock** contains information relative to the filesystem e.g. filesystem type, size, status, metadata structures. The **Master Boot Record (MBR)** is a 512-byte program located in the first sector of the hard disk; it contains information about hard disk partitions and has the duty of loading the OS. On recent systems, the MBR has been replaced by the **GUID Partition Table (GPT)**.

Almost all modern filesystems use **journaling**; in a journaling filesystem, the journal logs changes before committing them to the filesystem, which ensures faster recovery and less risk of corruption in case of a crash.

Partitioning limits for Linux using MBR:
- Max 4 primary partitions per hard disk, or 3 primary partitions + 1 extended partition. Partitions are numbered from 1 to 4.
- Max 11 logical partitions (inside the extended partition) per hard disk. Partitions are numbered from 5 to 15.
- Max disk size is 2 Tb.

GPT does not differentiate between primary, extended, or logical partitions. Furthermore, it practically has no limits concerning number and size of partitions.

**FUSE (Filesystem in Userspace)** is an interface for userspace programs to export a filesystem to the Linux kernel, and is particularly useful for virtual file systems.

```
fdisk /dev/sda  # Disk partitioning interactive tool
fdisk -l /dev/sda  # List the partition table of /dev/sda

parted  # Disk partitioning interactive tool
sfdisk /dev/sda  # Disk partitioning non-interactive tool
cfdisk  # Disk partitioning tool with text-based UI
gparted  # Disk partitioning tool with GUI
gnome-disks

partprobe device  # Notify the OS about partition table changes. Otherwise, the changes will take place only after reboot
hdparm -z device

blockdev --getbsz /dev/sdal  # Get the block size of the specified partition
```

The **UUID (Universal Unique Identifier)** of a partition is a 128-bit hash number, which is associated to the partition when the partition is initialized.

```
blkid /dev/sdal  # Print the UUID of the specified partition
blkid -L /boot  # Print the UUID of the specified partition, given its label
blkid -U 652b786e-b87f-49d2-af23-8087ced0c667  # Print the name of the specified partition, given its UUID

findfs UUID=652b786e-b87f-49d2-af23-8087ced0c667  # Print the name of the specified partition, given its UUID
findfs LABEL=/boot  # Print the name of the specified partition, given its label

e2label /dev/sdal  # Print the label of the specified partition
```
mkfs -t fstype device

Create a filesystem of the specified type on a partition (i.e. format the partition).
mkfs is a wrapper utility for the actual filesystem-specific maker commands:

- mkfs.ext2 aka mke2fs
- mkfs.ext3 aka mke3fs
- mkfs.ext4
- mkfs.msdos aka mkdosfs
- mkfs.ntfs aka mkntfs
- mkfs.reiserfs aka mkreiserfs
- mkfs.jfs
- mkfs.xfs

Example commands:

- mkfs -t ext2 /dev/sda
- mkfs.ext2 /dev/sda
- mke2fs /dev/sda
- mke2fs -j /dev/sda
- mkfs.ext3 /dev/sda
- mke3fs /dev/sda
- mkfs -t msdos /dev/sda
- mkfs.msdos /dev/sda
- mkdosfs /dev/sda

Create an ext2 filesystem on /dev/sda
Create an ext3 filesystem (ext2 with journaling) on /dev/sda
Create a MS-DOS filesystem on /dev/sda
mount

Display the currently mounted filesystems.

The commands `mount` and `umount` maintain in `/etc/mtab` a database of currently mounted filesystems, but `/proc/mounts` is authoritative

`mount -a`  
Mount all devices listed in `/etc/fstab`, except those indicated as `noauto`

`mount -t ext3 /dev/sda /mnt`  
Mount a Linux-formatted disk. The mount point (directory) must exist

`mount -t msdos /dev/fd0 /mnt`  
Mount a MS-DOS filesystem floppy disk to mount point `/mnt`

`mount /dev/fd0`  
Mount a floppy disk. `/etc/fstab` must contain an entry for `/dev/fd0`

`mount -o remount,rw /`  
Remount the root directory as read-write, supposing it was mounted read-only. Useful to change flags (in this case, read-only to read-write) for a mounted filesystem that cannot be unmounted at the moment

`mount -o nolock 10.7.7.7:/export/ /mnt/nfs`  
Mount a NFS share without running NFS daemons. Useful during system recovery

`mount -t iso9660 -o ro,loop=/dev/loop0 cd.img /mnt/cdrom`  
Mount a CD-ROM ISO9660 image file like a CD-ROM (via the loop device)

`umount /dev/fd0`  
Unmount a floppy disk that was mounted on `/mnt` (device must not be busy)

`umount /mnt`  
Unmount the floppy disk as soon as it is not in use anymore

`mountpoint /mnt`  
Tell if a directory is a mount point

`findmnt /dev/sda`  
List all mounted filesystems

`findmnt /mnt`  
Find the filesystem, given a specified device or mount point

`eject /dev/fd0`  
Eject a removable media device

`eject /mnt`  
Eject a removable media device
The command `sfdisk -T` prints the above list of partition IDs and names.

### Most used Linux-supported filesystems

- **ext2**: The oldest Linux ext filesystem, without journaling
- **ext3**: ext2 with journaling
- **ext4**: Linux journaling filesystem, an upgrade from ext3
- **Reiserfs**: Journaling filesystem
- **XFS**: Journaling filesystem, developed by SGI
- **JFS**: Journaling filesystem, developed by IBM
- **Btrfs**: B-tree filesystem, developed by Oracle
- **msdos**: DOS filesystem, supporting only 8-char filenames
- **umsdos**: Extended DOS filesystem used by Linux, compatible with DOS
- **fat32**: MS Windows FAT filesystem
- **vfat**: Extended DOS filesystem, with support for long filenames
- **ntfs**: Replacement for fat32 and vfat filesystems
- **minix**: Native filesystem of the MINIX OS
- **iso9660**: CD-ROM filesystem
- **cramfs**: Compressed RAM disk
- **nfs**: Network filesystem, used to access files on remote machines
- **SMB**: Server Message Block, used to mount MS Windows network shares
- **proc**: Pseudo filesystem, used as an interface to kernel data structures
- **swap**: Pseudo filesystem, Linux swap area
The swap space is an area on disk (a file or a partition) used as a RAM extension. When there is not enough free physical RAM for a process, inactive pages in memory are temporarily swapped out of memory to disk, to later be swapped in to memory when RAM resources are available again. If both RAM and swap space become nearly full, the system may get clogged by spending all the time paging blocks of memory back and forth between RAM and swap (thrashing). The amount of RAM plus the swap is defined as the virtual memory.

In Linux, a swap partition is usually preferred over a swap file. While a swap file can be resized more easily, it cannot be used for hibernation; this because the system must first locate the swap file's header, but in order to do so the filesystem containing the swap file must be mounted, and journaled filesystems such as ext3 or ext4 cannot be mounted during resume from disk. Also, in older Linux versions a swap partition used to have faster disk access and less fragmentation than a swap file, although the difference is negligible nowadays.

The swap partition is listed as filesystem type 0x82; however, it is not a filesystem, but a raw addressable memory space with no structure. For this reason it does not appear in the output of mount or df commands. A swap partition can be created via any partitioning tool e.g. fdisk.

```
dd if=/dev/zero of=/swapfile bs=1024 count=512000
mkswap /swapfile
swapon /swapfile
swapoff /swapfile
swapon -s
cat /proc/swaps
cat /proc/meminfo
free
top
```

Create a 512-Mb swap file

Initialize a (already created) swap file or partition

Enable a swap file or partition, thus telling the kernel that it can use it now

Disable a swap file or partition

Show the sizes of total and used swap areas

How to extend a LVM swap partition

1. lvs
   Determine the name of the swap Logical Volume
2. swapoff /dev/volgroup0/swap_lv
   Turn off the swap volume
3. lvresize -L+1G /dev/volgroup0/swap_lv
   Extend the swap volume with an additional 1 Gb of space
4. mkswap /dev/volgroup0/swap_lv
   Format the swap volume
5. swapon /dev/volgroup0/swap_lv
   Turn on the swap volume
### /etc/fstab

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Mount Point</th>
<th>Type</th>
<th>Options</th>
<th>Dump</th>
<th>Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda2</td>
<td>/</td>
<td>ext2</td>
<td>defaults</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>/dev/sdb1</td>
<td>/home</td>
<td>ext2</td>
<td>defaults</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>/dev/cdrom</td>
<td>/media/cdrom</td>
<td>auto</td>
<td>ro,noauto,user,exec</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/dev/fd0</td>
<td>/media/floppy</td>
<td>auto</td>
<td>rw,ro,noauto,user,async</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>proc</td>
<td>/proc</td>
<td>proc</td>
<td>defaults</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/dev/hdal</td>
<td>swap</td>
<td>swap</td>
<td>pri=42</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>nfsserver:/dirs</td>
<td>/mnt</td>
<td>nfs</td>
<td>intr</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>//smbserver/jdoe</td>
<td>/shares/jdoe</td>
<td>cifs</td>
<td>auto,credentials=/etc/smbcreds</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LABEL=/boot</td>
<td>/boot</td>
<td>ext4</td>
<td>errors=remount-ro,noatime</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>UUID=652b786e-b87f-49d2-af23-8087ced0c667</td>
<td>/test</td>
<td>ext4</td>
<td>errors=remount-ro,noatime</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

/`/etc/fstab` contains information about filesystems, including all filesystems that must be automatically mounted at bootup.

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Mount Point</th>
<th>Type</th>
<th>Options</th>
<th>Dump</th>
<th>Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ro</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>rw</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>suid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nosuid</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dev</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nodev</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>auto</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>noauto</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>user</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nouser</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>exec</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>noexec</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sync</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>async</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>noatime</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>acl</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>context=&quot;context&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other specific options apply to specific partition types (e.g. NFS or Samba)

**filesystem**

Device or partition. The filesystem can be identified either by its name, label, or UUID

**mount point**

Directory on which the partition will be mounted

**type**

Filesystem type, or **auto** if detected automatically

**options**

Use the default options. The default options depend on the filesystem type and can be found via the command:

tune2fs -l device | grep "Default mount options"

Most common default options: rw, suid, dev, auto, nouser, exec, async

**dump**

Options for the `dump` backup utility. 0 = do not backup

**pass**

Order in which the filesystem must be checked by `fsck`. 0 = do not check
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>df</td>
<td>Report filesystem disk space usage</td>
</tr>
<tr>
<td>df -h</td>
<td>Report filesystem disk space usage in human-readable output</td>
</tr>
<tr>
<td>df directory</td>
<td>Shows on which device the specified directory is mounted</td>
</tr>
<tr>
<td>du directory</td>
<td>Report disk usage, as the size of each file contained in directory, in Kb</td>
</tr>
<tr>
<td>du -s directory</td>
<td>Show the total sum of the sizes of all files contained in directory</td>
</tr>
<tr>
<td>du -h directory</td>
<td>Report disk usage in human-readable output</td>
</tr>
<tr>
<td>du -hs *</td>
<td>Sort -hr</td>
</tr>
<tr>
<td>du -a /path</td>
<td>Sort -nr</td>
</tr>
<tr>
<td>find /path -type f -exec du -Sh {} +</td>
<td>Sort -hr</td>
</tr>
<tr>
<td>ncdus</td>
<td>Disk usage analyzer with ncurses UI</td>
</tr>
<tr>
<td>resize2fs options device size</td>
<td>Resize an ext2/ext3/ext4 filesystem</td>
</tr>
<tr>
<td>lsblk</td>
<td>List information about all available block devices</td>
</tr>
<tr>
<td>lsscsi</td>
<td>List information about all SCSI devices</td>
</tr>
<tr>
<td>sync</td>
<td>Flush the buffer and commit all pending writes.</td>
</tr>
<tr>
<td>chroot /path/to/newrootdir command</td>
<td>Run a command in a chroot jail (i.e. in a new root directory). The command process will be unable to access files outside the chroot jail</td>
</tr>
<tr>
<td>chroot /mnt/sysimage</td>
<td>Start a shell with /mnt/sysimage as filesystem root. Useful during system recovery when the machine has been booted from a removable media; this device is defined as the filesystem root and often needs to be changed to perform operations on the machine</td>
</tr>
<tr>
<td>mknod /dev/ sda</td>
<td>Create a directory allocating the proper inode. Useful if experiencing filesystem problems during system recovery</td>
</tr>
<tr>
<td>multipath options device</td>
<td>Detect and aggregate multiple I/O paths (SAN connections) to a device</td>
</tr>
<tr>
<td>blkdiscard options device</td>
<td>Discard device sectors, wiping the data they contain. Useful for SSDs</td>
</tr>
<tr>
<td>hdparm</td>
<td>Get or set drive parameters for SATA/IDE devices</td>
</tr>
<tr>
<td>hdparm -g /dev/hda</td>
<td>Display drive geometry (cylinders, heads, sectors) of /dev/hda</td>
</tr>
<tr>
<td>hdparm -i /dev/hda</td>
<td>Display identification information for /dev/hda</td>
</tr>
<tr>
<td>hdparm -T /dev/hda</td>
<td>Perform disk read benchmarks on the /dev/hda drive</td>
</tr>
<tr>
<td>hdparm -p 12 /dev/hda</td>
<td>Reprogram IDE interface chipset of /dev/hda to mode 4. Warning: using an unsupported mode can cause filesystem corruption</td>
</tr>
<tr>
<td>sdparm</td>
<td>Access drive parameters for SCSI devices</td>
</tr>
</tbody>
</table>
Filesystem maintenance

**fsck device**

Check and repair a Linux filesystem (which must be unmounted). Corrupted files will be placed into the `/lost+found` directory of the partition. The exit code returned is the sum of the following conditions:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No errors</td>
</tr>
<tr>
<td>1</td>
<td>File system errors corrected</td>
</tr>
<tr>
<td>2</td>
<td>System should be rebooted</td>
</tr>
<tr>
<td>4</td>
<td>File system errors left uncorrected</td>
</tr>
<tr>
<td>8</td>
<td>Operational error</td>
</tr>
<tr>
<td>16</td>
<td>Usage or syntax error</td>
</tr>
<tr>
<td>32</td>
<td>Fsc canceled by user</td>
</tr>
<tr>
<td>128</td>
<td>Shared library error</td>
</tr>
</tbody>
</table>

Fsck is a wrapper utility for the actual filesystem-specific checker commands:

- `fsck.ext2` aka `e2fsck`
- `fsck.ext3`
- `fsck.ext4` aka `e2fsck`
- `fsck.msdos`
- `fsck.vfat`
- `fsck.cramfs`

**fsck**

- `fsck -A` Check and repair serially all filesystems listed in `/etc/fstab`
- `fsck -f /dev/sda1` Force a filesystem check on `/dev/sda1` even if it thinks is not necessary
- `fsck -y /dev/sda1` During filesystem repair, do not ask questions and assume that the answer is always yes
- `fsck.ext2 -c /dev/sda1` Check an ext2 filesystem, running the `badblocks` command to mark all bad blocks and add them to the bad block inode, so that they will not be allocated to files or directories
- `touch /forcefsck` (Red Hat) Force a filesystem check after next reboot

**tune2fs options device**

- `tune2fs -l /dev/sda1` List the contents of the filesystem superblock
- `tune2fs -j /dev/sda1` Add a journal to this ext2 filesystem, making it an ext3
- `tune2fs -m 1 /dev/sda1` Reserve 1% of the partition size to privileged processes. This space (5% by default, but can be reduced on modern filesystems) is reserved to avoid filesystem fragmentation and to allow privileged processes to continue to run correctly when the partition is full
- `tune2fs -C 7 /dev/sda1` Set the mount count of the filesystem to 7
- `tune2fs -c 20 /dev/sda1` Set the filesystem to be checked by fsck after 20 mounts
- `tune2fs -i 15d /dev/sda1` Set the filesystem to be checked by fsck each 15 days

Both mount-count-dependent and time-dependent checking are enabled by default for all hard drives on Linux, to avoid the risk of filesystem corruption going unnoticed.

**dumpe2fs options device**

- `dumpe2fs -h /dev/sda1` Dump ext2/ext3/ext4 filesystem information
- `dumpe2fs -i superblock` Display filesystem’s superblock information (number of mounts, last checks, UUID, etc.)
- `dumpe2fs /dev/sda1 | grep -i superblock` Display locations of superblock (primary and backup) of filesystem
- `dumpe2fs -b /dev/sda1` Display blocks that are marked as bad in the filesystem

**debugfs options device**

- `debugfs -w /dev/sda1` Interactive ext2/ext3/ext4 filesystem debugger
- `debugfs -w /dev/sda1` Debug /dev/sda1 in read-write mode (by default, debugfs accesses the device in read-only mode)

Many hard drives feature the **Self-Monitoring, Analysis and Reporting Technology (SMART)** whose purpose is to monitor the reliability of the drive, predict drive failures, and carry out different types of drive self-tests. The `smartctl` daemon attempts to poll this information from all drives every 30 minutes, logging all data to syslog.

- `smartctl -a /dev/sda` Print SMART information for drive /dev/sda
- `smartctl -s off /dev/sda` Disable SMART monitoring and log collection for drive /dev/sda
- `smartctl -t long /dev/sda` Begin an extended SMART self-test on drive /dev/sda
XFS, ReiserFS, and CD-ROM filesystems

xfs_growfs options mountpoint
Expand an XFS filesystem.
The opposite operation is not possible: a XFS filesystem cannot be shrunk
xfs_info /dev/sdal
Print XFS filesystem geometry
xfs_growfs -n /dev/sdal
xfs_check options device
Check XFS filesystem consistency
xfs_repair options device
Repair a damaged or corrupt XFS filesystem
xfsdump -v silent -f /dev/tape /
Dump the root of a XFS filesystem to tape, with the lowest verbosity.
Incremental and resumed dumps are stored in the inventory database
/var/lib/xfsdump/inventory
xfsrestore -f /dev/tape /
Restore a XFS filesystem from tape
xfsdump -J - / | xfsrestore -J - /new
Copy the contents of a XFS filesystem to another directory, without updating the inventory database

reiserfstune options device
Adjust tunable filesystem parameters on ReiserFS filesystem
debugreiserfs device
Interactive ReiserFS filesystem debugger

mkisofs -r -o cdrom.img data/
Create a CD-ROM image from the contents of the target directory.
Enables Rock Ridge extension and sets all content on CD to be publicly readable, instead of inheriting the permissions from the original files

### CD-ROM filesystems

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO9660</td>
<td>mkisofs</td>
<td>Create a ISO9660 filesystem</td>
</tr>
<tr>
<td>UDF (Universal Disk Format)</td>
<td>mkudfss</td>
<td>Create a UDF filesystem</td>
</tr>
<tr>
<td></td>
<td>udfssck</td>
<td>Check a UDF filesystem</td>
</tr>
<tr>
<td></td>
<td>wrudf</td>
<td>Maintain a UDF filesystem</td>
</tr>
<tr>
<td></td>
<td>cdrwtool</td>
<td>Manage CD-RW drives (e.g. disk format, read/write speed)</td>
</tr>
<tr>
<td>HFS (Hierarchical File System)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CD-ROM filesystem extensions

| Rock Ridge                    | Contains the original file information (e.g. permissions, filename) for MS Windows 8.3 filenames |
| MS Joliet                     | Used to create more MS Windows friendly CD-ROMs                                               |
| El Torito                     | Used to create bootable CD-ROMs                                                                  |
AutoFS is a client-side service that allows automounting of filesystems, even for nonprivileged users. AutoFS is composed of the `autofs` kernel module that monitors specific directories for attempts to access them; in this case, the kernel module signals the `automount` userspace daemon, which mounts the directory when it needs to be accessed and unmounts it when it is no longer accessed. Mounts managed by AutoFS should not be mounted/unmounted manually or via `/etc/fstab`, to avoid inconsistencies.

### AutoFS configuration files

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/sysconfig/autofs</code></td>
<td>AutoFS configuration file.</td>
</tr>
<tr>
<td><code>/etc/auto.master</code></td>
<td>Master map file for AutoFS. Each line is an indirect map, and each map file stores the configuration for the automounting of the subdirectory. The <code>-hosts</code> map tells AutoFS to mount/unmount automatically any export from the NFS server <code>nfsserver</code> when the directory <code>/net/nfsserver/</code> is accessed.</td>
</tr>
</tbody>
</table>

```plaintext
# mount point   map              options
/net            -hosts
/-              /etc/auto.direct
/misc           /etc/auto.misc
/home           /etc/auto.home --timeout=60
```

### AutoFS map files

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/auto.direct</code></td>
<td>Direct map file for automounting of a NFS share.</td>
</tr>
</tbody>
</table>

```plaintext
# dir      filesystem
/mydir     nfsserver1.foo.org:/myshare
```

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/auto.misc</code></td>
<td>Indirect map file for automounting of directory <code>/misc</code>.</td>
</tr>
</tbody>
</table>

```plaintext
# subdir   options                          filesystem
/public    -ro,soft,intr                    ftp.example.org:/pub
/cd        -fstype=iso9660,ro,nosuid,nodev  :/dev/cdrom
```

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/auto.home</code></td>
<td>Indirect map file for automounting of directory <code>/home</code> on a NFS share. The <code>*</code> wildcard matches any subdirectory the system attempts to access, and the <code>&amp;</code> variable takes the value of the match.</td>
</tr>
</tbody>
</table>

```plaintext
# subdir   options                          filesystem
/*          -rw,soft,intr                    nfsserver2.bar.org:/home/`
```
<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Storage capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 0</td>
<td>Striping (data is written across all member disks). High I/O but no redundancy</td>
<td>Sum of the capacity of member disks</td>
</tr>
<tr>
<td>RAID 1</td>
<td>Mirroring (data is mirrored on all disks). High redundancy but high cost</td>
<td>Capacity of the smaller member disk</td>
</tr>
<tr>
<td>RAID 4</td>
<td>Parity on a single disk. I/O bottleneck unless coupled to write-back caching</td>
<td>Sum of the capacity of member disks, minus one</td>
</tr>
<tr>
<td>RAID 5</td>
<td>Parity distributed across all disks. Can sustain one disk crash</td>
<td>Sum of the capacity of member disks, minus one</td>
</tr>
<tr>
<td>RAID 6</td>
<td>Double parity distributed across all disks. Can sustain two disk crashes</td>
<td>Sum of the capacity of member disks, minus two</td>
</tr>
<tr>
<td>RAID 10 (1+0)</td>
<td>Striping + mirroring. High redundancy but high cost</td>
<td>Capacity of the smaller member disk</td>
</tr>
<tr>
<td>Linear RAID</td>
<td>Data written sequentially across all disks. No redundancy</td>
<td>Sum of the capacity of member disks</td>
</tr>
</tbody>
</table>

```
mdadm -C /dev/md0 -l 5 -n 3 /dev/sdb1 /dev/sdc1 /dev/sdd1 -x 1 /dev/sde1
```

Create a RAID 5 array from three partitions and a spare. Partitions type must be set to 0xFD. Once the RAID device has been created, it must be formatted e.g. via `mke2fs -j /dev/md0`

```
mdadm --manage /dev/md0 -f /dev/sdd1
mdadm --manage /dev/md0 -r /dev/sdd1
mdadm --manage /dev/md0 -a /dev/sdd1
```

Mark a drive as faulty, before removing it
Remove a drive from the RAID array. The faulty drive can then be physically removed
Add a drive to the RAID array. To be run after the faulty drive has been physically replaced

```
mdadm --misc -Q /dev/sdd1
mdadm --misc -D /dev/md0
mdadm --misc -o /dev/md0
mdadm --misc -w /dev/md0
```

Display information about a device
Display detailed information about the RAID array
Mark the RAID array as readonly
Mark the RAID array as read & write

/etc/mdadm.conf

Configuration file for the `mdadm` command

```
DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
ARRAY /dev/md0 level=raid5 num-devices=3
UUID=0098af43:812203fa:e665b421:002f5e42
devices=/dev/sdb1, /dev/sdc1, /dev/sdd1, /dev/sde1
```

```
cat /proc/mdstat
```

Display information about RAID arrays and devices
<table>
<thead>
<tr>
<th>Non-GRUB bootloaders</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LILO</strong> (Linux Loader)</td>
</tr>
<tr>
<td><strong>SYSLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>ISOLINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>PXELINUX</strong></td>
</tr>
<tr>
<td><strong>EXTLINUX</strong></td>
</tr>
</tbody>
</table>
GRUB (Grand Unified Bootloader) is the standard boot manager on Linux distributions. The latest version is GRUB 2; the older version is GRUB Legacy.

GRUB Stage 1 (446 bytes), as well as the partition table (64 bytes) and the boot signature (2 bytes), is stored in the 512-byte MBR. It then accesses the GRUB configuration and commands available on the filesystem, usually on /boot/grub.

```
# /boot/grub/grub.cfg or /boot/grub2/grub.cfg  GRUB 2 configuration file

# Linux Red Hat
menuentry "Fedora 2.6.32" {    # Menu item to show on GRUB bootmenu
  set root=(hd0,1)              # root filesystem is /dev/hda1
  linux /vmlinuz-2.6.32 ro root=/dev/hda5 mem=2048M
  initrd /initrd-2.6.32
}

# Linux Debian
menuentry "Debian 2.6.36-experimental" {
  set root=(hd0,1)
  linux (hd0,1)/bzImage-2.6.36-experimental ro root=/dev/hda6
}

# MS Windows
menuentry "MS Windows" {
  set root=(hd0,2)
  chainloader +1
}
```

The GRUB 2 configuration file must not be edited manually. Instead, it is necessary to edit the files in /etc/grub.d/ (scripts that will be run in order) and the file /etc/default/grub (configuration file for menu display settings), then run update-grub (Debian) or grub2-mkconfig (Red Hat) which will recreate this configuration file.

**Common kernel parameters:**

- `root=` Specify the location of the filesystem root. This is a required parameter
- `ro` Mount read-only on boot
- `quiet` Disable non-critical kernel messages during boot
- `debug` Enable kernel debugging
- `splash` Show splash image
- `single` Boot in single-user mode (runlevel 1)
- `emergency` Emergency mode: after the kernel is booted, run `sulogin` (single-user login) which asks for the root password for system maintenance, then run a Bash shell. Does not load init or any daemon or configuration setting
- `init=/bin/bash` Run a Bash shell (may also be any other executable) instead of init
The GRUB menu, presented at startup, allows to choose the OS or kernel to boot:
- **ENTER**: Boot the currently selected GRUB entry
- **C**: Get a GRUB command line
- **E**: Edit the selected GRUB entry (e.g. to edit kernel parameters in order to boot in single-user emergency mode, or to change IRQ or I/O port of a device driver compiled in the kernel)
- **B**: Boot the currently selected GRUB entry. This is usually done after finishing modifying the entry
- **P**: Bring up the GRUB password prompt. Necessary if a GRUB password has been set

```
grub2-mkconfig -o /boot/grub2/grub.cfg (BIOS)  # Regenerate GRUB configuration file
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg (EFI)
grub-install /dev/sda                  # Install GRUB on first SATA drive
grub                                      # Access the GRUB shell
grub2-set-default 1                       # Set GRUB to automatically boot the second entry in the GRUB menu
grub2-editenv list                        # Display the current GRUB menu entry that is automatically booted

/boot/grub/device.map
```

This file can be created to map Linux device filenames to BIOS drives:

```
(fd0)  /dev/fd0
(hd0)  /dev/hda
```
GRUB Legacy - configuration

```
timeout 10   # Boot the default kernel after 10 seconds
default 0    # Default kernel is 0

# Section 0: Linux boot
title   Debian    # Menu item to show on GRUB bootmenu
root    (hd0,0)   # root filesystem is /dev/hda1
kernel /boot/vmlinuz-2.6.24-19-generic root=/dev/hda1 ro quiet splash
initrd /boot/initrd.img-2.6.24-19-generic

# Section 1: MS Windows boot
title   Microsoft Windows XP
root    (hd0,1)   # root filesystem is /dev/hda2
savedefault
makeactive        # set the active flag on this partition
chainloader +1    # read 1 sector from start of partition and run

# Section 2: Firmware/BIOS update from floppy disk
title   Firmware update
kernel /memdisk   # boot a floppy disk image
initrd /floppy-img-7.7.7
```
### GRUB Legacy shell commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blocklist file</td>
<td>Print the block list notation of a file</td>
</tr>
<tr>
<td>boot</td>
<td>Boot the loaded OS</td>
</tr>
<tr>
<td>cat file</td>
<td>Show the contents of a file</td>
</tr>
<tr>
<td>chainloader file</td>
<td>Chainload another bootloader</td>
</tr>
<tr>
<td>cmp file1 file2</td>
<td>Compare two files</td>
</tr>
<tr>
<td>configfile file</td>
<td>Load a configuration file</td>
</tr>
<tr>
<td>debug</td>
<td>Toggle debugging mode</td>
</tr>
<tr>
<td>displayapm</td>
<td>Display APM BIOS information</td>
</tr>
<tr>
<td>displaymem</td>
<td>Display memory configuration</td>
</tr>
<tr>
<td>embed stage device</td>
<td>Embed Stage 1.5 in the device</td>
</tr>
<tr>
<td>find file</td>
<td>Find a file</td>
</tr>
<tr>
<td>ftest</td>
<td>Toggle filesystem test mode</td>
</tr>
<tr>
<td>geometry drive</td>
<td>Print information on a drive geometry</td>
</tr>
<tr>
<td>halt</td>
<td>Shut down the system</td>
</tr>
<tr>
<td>help command</td>
<td>Show help for a command, or the available commands</td>
</tr>
<tr>
<td>impsprobe</td>
<td>Probe the Intel Multiprocessor Specification</td>
</tr>
<tr>
<td>initrd file</td>
<td>Load an initial ramdisk image file</td>
</tr>
<tr>
<td>install options</td>
<td>Install GRUB (deprecated; setup should be used instead)</td>
</tr>
<tr>
<td>ioprobe drive</td>
<td>Probe I/O ports used for a drive</td>
</tr>
<tr>
<td>kernel file</td>
<td>Load a kernel</td>
</tr>
<tr>
<td>lock</td>
<td>Lock a GRUB menu entry</td>
</tr>
<tr>
<td>makeactive</td>
<td>Set active partition on root disk to GRUB’s root device</td>
</tr>
<tr>
<td>map drive1 drive2</td>
<td>Map a drive to another drive</td>
</tr>
<tr>
<td>md5crypt</td>
<td>Encrypt a password in MD5 format</td>
</tr>
<tr>
<td>module file</td>
<td>Load a kernel module</td>
</tr>
<tr>
<td>modulenounzip file</td>
<td>Load a kernel module without decompressing it</td>
</tr>
<tr>
<td>pause message</td>
<td>Print a message and wait for a key press</td>
</tr>
<tr>
<td>quit</td>
<td>Quit the GRUB shell</td>
</tr>
<tr>
<td>reboot</td>
<td>Reboot the system</td>
</tr>
<tr>
<td>read address</td>
<td>Read a 32-bit value from memory and print it</td>
</tr>
<tr>
<td>root device</td>
<td>Set the current root device</td>
</tr>
<tr>
<td>rootnoverify device</td>
<td>Set the current root device without mounting it</td>
</tr>
<tr>
<td>savedefault</td>
<td>Save current menu entry as the default entry</td>
</tr>
<tr>
<td>setup device</td>
<td>Install GRUB automatically on the device</td>
</tr>
<tr>
<td>testload file</td>
<td>Test the filesystem code on a file</td>
</tr>
<tr>
<td>testvbe mode</td>
<td>Test a VESA BIOS EXTENSION mode</td>
</tr>
<tr>
<td>uppermem kbytes</td>
<td>Set the upper memory size (only for old machines)</td>
</tr>
<tr>
<td>vbeprobe mode</td>
<td>Probe a VESA BIOS EXTENSION mode</td>
</tr>
</tbody>
</table>
dpkg is the low-level package manager for Debian. It uses the DEB package format, which is compressed with ar.

- `dpkg -i package.deb`: Install a package file
- `dpkg -r package`: Remove a package
- `dpkg -l`: List installed packages and their state
- `dpkg -L package`: List the content of an installed package
- `dpkg -c package.deb`: List the content of a package file
- `dpkg -S file`: Show the package containing a specific file
- `dpkg-reconfigure package`: Reconfigure a package

apt is the high-level package manager for Debian.

High-level package managers are able to install remote packages and automatically solve dependencies.

- `apt-get install package`: Install a package
- `apt-get remove package`: Remove a package
- `apt-get upgrade`: Upgrade all installed packages
- `apt-get dist-upgrade`: Upgrade all installed packages and handle dependencies with new versions
- `apt-get source package`: Get the source code for a package
- `apt-get check`: Check for broken dependencies and update package cache
- `apt-get install -f`: Fix broken dependencies
- `apt-get update`: Update information on available packages
- `apt-get search package`: Search for a package
- `apt-cache depends package`: Show package dependencies
- `apt-cache show package`: Show package records
- `apt-cache showpkg package`: Show information about a package
- `apt-file update`: Update information about package contents
- `apt-file list package`: List the content of an uninstalled package
- `apt-file search file`: Show which package provides a specific file
- `apt-key add keyfile`: Add a key to the list of keys used to authenticate packages
- `apt-cdrom add`: Add a CD-ROM to the sources list
- `cat /etc/apt/sources.list`: Print list of available repositories
- `alien -i package.rpm`: Convert an RPM package to DEB and install it.
  Warning: might break the package database system

Package manager with text interface, front-end to dpkg. Obsolete

- `dselect`

Package manager with ncurses UI, front-end to apt

- `aptitude`

Package manager with Gtk+ UI, front-end to apt

- `synaptic`
rpm is the low-level package manager for Red Hat. It uses the RPM package format, which is cpio-compressed.

```
rpm -i package.rpm  # Install a package file
rpm -i ftp://host/package.rpm
rpm -i http://host/package.rpm
rpm -e package  # Remove a package
rpm -U package.rpm  # Upgrade a package (and remove old versions)
rpm -F package.rpm  # Upgrade a package (only if an old version is already installed)
rpm -qa  # List installed packages and their state
rpm -qa --last  # List installed packages and their installation date, from newest to oldest
rpm -ql package  # List the content of an installed package
rpm -qpl package.rpm  # List the content of a package file
rpm -qf file  # Show the package containing a specific file
rpm -V package  # Verify an installed package
rpm -i package.src.rpm  # Install a package source file
rpm -ba package.spec  # Compile a package source file
rpm2cpio package.rpm  # Convert an RPM package to a cpio archive
creatererepo directory  # Create an XML file of repository metadata from the set of RPMs contained in directory

pirut  # Package manager with GUI. Obsolete
```
yum is the high-level package manager for Red Hat up to RHEL 7. In RHEL 8, it is a front-end to dnf.

yum install package  
Install a package

yum install package.rpm  
Install a package file

yum localinstall package.rpm  
Remove a package

yum remove package  
Update an installed package

yum update package  
Update all installed packages

yum update  
Update all installed packages and remove obsolete packages

yum upgrade  
Replace a package with another

yum list  
List all installed and available packages

yum list searchterm  
List installed and available packages matching the search term

yum list installed  
List installed packages

yum list available  
List packages available for install

yum search searchterm  
Search for packages that match the search term in the package name or summary

yum search all searchterm  
Search for packages that match the search term in the package name, summary, or description

yum deplist package  
Show package dependencies (recursively)

yum list package  
Show package records

yum info package  
Show information about a package

yum history  
Show the installation history (installs, updates, etc.)

yum history list  
Show item n of the installation history

yum history info n  
Show detailed information on item n of the installation history (begin and end times, packages altered, etc.)

yum history package package  
Show the installation history about a package

yum whatprovides file  
Show which package provides a specific file

yum cmd --disablerepo="*" --enablerepo="repo"  
Execute the yum command but only with a specific repository repo enabled

yum repolist  
Print list of available repositories

cat /etc/yum.repos.d/*repo  
Delete temporary files for repositories

yum clean all  
rmdir /var/cache/yum

yumdownloader --resolve package  
Download package and all its dependencies

yumdownloader --urls package  
Show URLs that would be downloaded

yum-complete-transaction  
Try to complete unfinished or aborted package installations

repoquery --tree-requires package  
Show a tree with all dependencies of package
### Configuration of a Fedora repository (Red Hat)

<table>
<thead>
<tr>
<th>Repository ID</th>
<th>Repository name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[fedora]</td>
<td>name=Fedora $releasever - $basearch</td>
</tr>
</tbody>
</table>
|               | baseurl=http://download.fedoraproject.org/pub/fedora/
|               |   linux/releases/$releasever/Everything/$basearch/os/
|               |   http://foo.org/linux/$releasever/$basearch/os/
|               |   http://bar.org/linux/$releasever/$basearch/os/ |

- **enabled=1**
- **gpgcheck=1**
- **failovermethod=priority**

**List of URLs to the repository’s repodata directory.** Can be any of these types:
- **file:///** local file
- **file://** NFS
- **http://** HTTP
- **https://** HTTPS
- **ftp://** FTP

- **metalink=https://mirrors.fedoraproject.org/metalink?repo=fedora-$releasever&arch=$basearch**

- **gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch**

This repository configuration must be located in a repo file e.g. `/etc/yum.repos.d/fedora.repo`. The same repo file can contain multiple repository definitions.

The manpage `man yum.conf` lists all repository configuration options.

---

### How to install a package on an offline machine

The problem of installing a package on an offline machine is that the machine is unable to download the package dependencies. To solve this problem, first create an online machine identical to the offline machine, and with the smallest possible set of packages installed. Then proceed as follows.

**On the online machine:**

1. Install the package and all its dependencies in a local directory
   ```bash
   mkdir /tmp/repo
   yum --downloadonly --downloaddir=/tmp/repo install package
   ```

2. Create a local yum repository
   ```bash
   createrepo /tmp/repo
   chmod -R root:root /tmp/repo
   chown -R root:root /tmp/repo
   ```

3. Transfer the directory `/tmp/repo` from the online machine to the offline machine

**On the offline machine:**

4. Create a yum repo file `/etc/yum.repos.d/local.repo` for the new repository
   ```bash
   [local]
   name=Local
   baseurl=file:///tmp/repo
   enabled=1
   gpgcheck=0
   protect=1
   ```

5. Install the package from the local repository
   ```bash
   yum install package
   ```
Backup

**dd**

Tool to copy data, byte by byte, from a file or block device. Should not be used on a mounted block device, because of write cache issues.

```
 dd if=/dev/sda of=/dev/sdb
```

Copy the content of one hard disk over another

```
cat /dev/sda > /dev/sdb
```

Generate the image file of a partition

```
dd if=/dev/cdrom of=cdrom.iso bs=2048
```

Create an ISO file from a CD-ROM, using a block size transfer of 2 Kb

```
dd if=install.iso of=/dev/sdc bs=512k
```

Write an installation ISO file to a device (e.g. a USB thumb drive)

**ddrescue**

Data recovery tool. Like `dd`, but with high tolerance for read errors

**testdisk**

Data recovery tool. Recovers data from a deleted or corrupted partition

**photorec**

Data recovery tool. Recovers graphical image and video files from media such as digital cameras and CD-ROMs

**extundelete**

Data recovery tool. Recovers data from a EXT3 or EXT4 partition

**ntfsundelete**

Data recovery tool. Recovers data from a NTFS partition

**scalpel**

Data recovery tool. Recovers data from a disk image or a raw block device

**rsync**

Tool for local and remote file synchronization. For all copies after the first, copies only the blocks that have changed, making it a very fast and bandwidth-efficient backup solution

```
rsync -rzv /home  /tmp/bak
```

Synchronize the content of the home directory with the temporary backup directory; use recursion, compression, and verbosity

```
rsync -rzv /home/ /tmp/bak/home
```

Synchronize the content of the home directory with the backup directory on the remote server via SSH; use archive mode (operates recursively and preserves owner, group, permissions, timestamps, and symlinks)

```
rsync -avz /home root@10.0.0.7:/backup/
```

**burp**

Backup and restore program

---

### Tape libraries

<table>
<thead>
<tr>
<th>Devices</th>
<th>First SCSI tape device</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/st0</td>
<td>First SCSI tape device (no-rewind device file)</td>
</tr>
<tr>
<td>/dev/nst0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utility for magnetic tapes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mt -f /dev/nst0 asf 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utility for tape libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>mt -f /dev/sgl status</td>
</tr>
<tr>
<td>mt -f /dev/sgl load 3</td>
</tr>
<tr>
<td>mt -f /dev/sgl unload</td>
</tr>
<tr>
<td>mt -f /dev/sgl transfer 3 4</td>
</tr>
<tr>
<td>mt -f /dev/sgl inventory</td>
</tr>
<tr>
<td>mt -f /dev/sgl inquiry</td>
</tr>
<tr>
<td>Format</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td><strong>cpio</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>gzip</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>bzip2</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>7-Zip</strong></td>
</tr>
<tr>
<td><strong>xz</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>LZMA</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>rar</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>tar</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>star</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Command</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td><code>man command</code></td>
</tr>
<tr>
<td><code>man n command</code></td>
</tr>
<tr>
<td><code>man man</code></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><code>man n intro</code></td>
</tr>
<tr>
<td><code>mandb</code></td>
</tr>
<tr>
<td><code>whatis command</code></td>
</tr>
<tr>
<td><code>apropos keyword</code></td>
</tr>
<tr>
<td><code>man -k keyword</code></td>
</tr>
<tr>
<td><code>apropos -r regex</code></td>
</tr>
<tr>
<td><code>man -r regex</code></td>
</tr>
<tr>
<td><code>info command</code></td>
</tr>
<tr>
<td><code>help</code></td>
</tr>
<tr>
<td><code>help command</code></td>
</tr>
</tbody>
</table>

```
yum whatprovides /usr/share/man/man/n/command.n.gz (Red Hat)  # Find which package provides section `n` of the `command` manpage
yum install man-pages (Red Hat)                               # Install a large number of manpages from the Linux Documentation Project
yum install man-db (Red Hat)                                 # Install various manpage commands and utilities
```
Text filters

```
cat file
```
Print a text file

```
cat file1 file2 > file3
```
Concatenate text files

```
cat file1 > file2
> file2 < file1 cat
```
Copy file1 to file2. The cat command is able to operate on binary streams as well and therefore this command also works with binary files (e.g. JPG images)

```
cat > file <<EOF
line 1
line 2
line 3
EOF
```
Create a Here Document, storing the lines entered in input to file. EOF can be any text

```
command <<< 'string'
cat -etv <<< 'string'
```
Create a Here String, passing string as input to command

Print string, showing all invisible characters

```
tac file
```
Print or concatenate text files in opposite order line-wise, from last line to first line

```
rev file
```
Print a text file with every line reversed character-wise, from last char to first char

```
head file
head -n 10 file
```
Print the first 10 lines of a text file

```
tail file
tail -n 10 file
tail -f file
tail -n +1 file1 file2 file3
```
Print the last 10 lines of a text file

Output appended data as the text file grows. Useful to read a logfile in real-time

Print each file with a filename header

```
tail -f
```
tail for multiple files at the same time (ncurses UI)

```
column file
col file
```
Format a text file into columns

```
pr file
```
Format a text file for a printer

```
fmt -w 75 file
fold -w40 file
```
Format a text file so that each line has a max width of 75 characters

Wrap each line of a text file to 40 characters

Prepend line numbers to a text file

Print the number of lines, words, and bytes of a text file

```
join file1 file2
paste file1 file2
split -l 1 file
uniq file
sort file
shuf file
```
Join lines of two text files on a common field

Merge lines of text files

Split a text file into 1-line files; these will be named xaa, xab, xac, etc.

Print the unique lines of a text file, omitting consecutive identical lines

Sort alphabetically the lines of a text file

Shuffle randomly the lines of a text file

```
expand file
unexpand file
```
Convert tabs into spaces

Convert spaces into tabs

```
diff file1 file2
cmp file1 file2
```
Compare two text files line by line and print the differences

Compare two binary files byte by byte and print the differences
Advanced text filters

```bash
cut -d: -f3 file

cut -d: -f1 /etc/passwd

cut -c3-50 file

sed 's/foo/bar/' file

sed -i 's/foo/bar/g' file

sed '0,/foo/s//bar/' file

sed 's/foo/$var/' file

tr a-z A-Z <file

tr [:lower:] [:upper:] <file

tr -d 0-9 <file

tr -d [:digit:] <file

awk

rpl oldstring newstring file

grep foo file

grep -v foo file

grep -e foo -e bar file

grep -E 'foo|bar' file

grep -V file

egrep regex file

tail -f file | grep --line-buffered foo

stdbuf option command

tidy

tidy -asxml -xml -indent -wrap 2000 \ -quiet --hide-comments yes file.xml

json_verify < file.json

json_reformat < file.json

strings file

antiword file.doc

catdoc file.doc
```

Cut the lines of a file, considering : as the delimiter and printing only the 3rd field

Print the list of local user accounts in the system

Print character 3 to 50 of each line of a file

Stream Editor: Replace the first occurrence on a line of “foo” with “bar” in file, and print on stdout the result

Replace “foo” with “bar”, overwriting the results in file

Replace all occurrences of “foo” with “bar”

Replace only the first line match

Print line 7 to 13 of a text file

Replace “foo” with the value of variable $var. The double quotes are necessary for variable expansion

Translate characters: Convert all lowercase into uppercase in a text file

Delete all digits from a text file

Interpreter for the AWK programming language, designed for text processing and data extraction

Replace strings in a file

Print the lines of a file containing "foo"

Print the lines of a file not containing "foo"

Print the lines of a file containing "foo" or "bar"

Print the lines of a file containing neither "foo" nor "bar"

Output appended data as the text file grows, printing only the lines containing "foo"

Run command with modified stdin, stdout, or stderr buffering

Correct and tidy up the markup of HTML, XHTML, and XML files

Strip out comments from an XML file

Validate the syntax of a JSON file

Pretty format a JSON file

Show all printable character sequences at least 4-characters long that are contained in file

Show text and images from a MS Word document

Output plaintext from a MS Word document
Regular expressions

^  Beginning of a line
$  End of a line
\< \>  Word boundaries (beginning of line, end of line, space, or punctuation mark)
.  Any character except newline
[abc]  Any of the characters specified
[a-z]  Any of the characters in the specified range
[^abc]  Any character except those specified
*  Zero or more times the preceding regex
+  One or more times the preceding regex
?  Zero or one time the preceding regex
{5}  Exactly 5 times the preceding regex
{5,}  5 times or more the preceding regex
{5,10}  At most 10 times the preceding regex
{5,10}  Between 5 and 10 times the preceding regex
( )  Grouping, to be used for back-references. \1 expands to the 1st match, \2 to the 2nd, etc. up to \9

The symbols above are used in POSIX EREs (Extended Regular Expressions).
In POSIX BREs (Basic Regular Expressions), the symbols \ ? + { | ( ) need to be escaped; this is done by adding a backslash character \ in front of them.
### File management

- **cp file file2**  
  Copy a file

- **cp file dir/**  
  Copy a file to a directory

- **cp -ar /dir1/. /dir2/**  
  Copy a directory recursively

- **mv file file2**  
  Rename a file

- **mv file dir/**  
  Move a file to a directory

- **rm file**  
  Delete a file

- **pv file > file2**  
  Copy a file, monitoring the progress of data through a pipe

- **rename str1 str2 file**  
  Rename a file, replacing in the filename the first occurrence of string `str1` with `str2`

- **unlink file**  
  Remove the hard link to a file (equivalent to `rm`)

- **touch file**  
  Change access timestamp and modify timestamp of a file as now. If the file does not exist, it is created

- **truncate -s size file**  
  Shrink or extend a file to the specified size. If the file is larger than the specified size, it is truncated; if the file is shorter, the extra space is filled with zeros

- **mktemp**  
  Create a temporary file or directory, using `tmp.XXXXXXXXXX` as filename template

- **fdupes dir**  
  Examine a directory for duplicate files in it. To consider files a duplicate, it first compares file sizes and MD5 signatures, then file contents byte-by-byte

- **shred /dev/hda**  
  Securely wipe the contents of a device

- **shred -u file**  
  Securely delete a file

---

#### File-naming wildcards (globbing)

- `*`  
  Matches zero or more characters

- `?`  
  Matches one character

- `[abc]`  
  Matches a, b, or c

- `![abc]`  
  Matches any character except a, b, or c

- `[a-z]`  
  Matches any character between a and z

---

#### Brace expansion

- **cp foo.{txt,bak}**  
  Copy file "foo.txt" to "foo.bak"

- **touch foo_{a,b,c}**  
  Create files "foo_a", "foo_b", "foo_c"

- **touch foo_{a..c}**  
  Create files "foo_a", "foo_b", "foo_c"
cd directory Change to the specified directory

cd - Change to the previously used directory

pwd Print the current working directory

ls List the contents of the current directory

dir

ls -d */ List only directories contained on the current directory

ls -lap --sort=v List files, sorted by version number

mkdir dir Create a directory

mkdir -m 755 dir Create a directory with mode 755

mkdir -p/dir1/dir2/dir3 Create a directory, creating also the parent directories if they don't exist

rmdir dir Delete a directory (which must be empty)

tree List directories and their contents in hierarchical format

dirs Display the directory stack (i.e. the list of remembered directories)

pushd dir Add dir to the top of the directory stack and make it the current working directory

popd Remove the top directory from the directory stack and change to the new top directory

dirname file Output the directory path in which file is located, stripping any non-directory suffix from the filename

realpath file Output the resolved absolute path of file

### Bash directory shortcuts

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Current directory</td>
</tr>
<tr>
<td>..</td>
<td>Parent directory</td>
</tr>
<tr>
<td>~</td>
<td>Home directory of current user</td>
</tr>
<tr>
<td>~user</td>
<td>Home directory of user</td>
</tr>
<tr>
<td>~-</td>
<td>Previously used directory</td>
</tr>
</tbody>
</table>
File access and encoding

**`lsomal`**  
List all open files

**`lsomal -u user`**  
List all files currently open by *user*

**`lsomal -i`**  
List open files and their sockets (equivalent to `netstat -ap`)

**`lsomal -i :80`**  
List connections of local processes on port 80

**`lsomal -i@10.0.0.3`**  
List connections of local processes to remote host 10.0.0.3

**`lsomal -i@10.0.0.3:80`**  
List connections of local processes to remote host 10.0.0.3 on port 80

**`lsomal -c mysql`**  
List all files opened by `mysql`, the MySQL daemon

**`lsomal file`**  
List all processes using a specific *file*

**`lsomal +L1`**  
List open files with a link count of 0 i.e. that have been unlinked. These files are not accessible but take up disk space. A process holding such a file prevents the system from deleting it (thus freeing disk space), until the process is killed or restarted

**`fuser`**  
Show the name of processes using a specific file, directory, or socket

**`fuser -v file`**  
Show the name of the process using *file*

**`fuser -v -n tcp 443`**  
Show the name of the process running on port 443

**`lslocks`**  
List information about all currently held file locks

**`tmpwatch`**  
Remove files which have not been accessed for some time

**`stat file`**  
Display file or filesystem status

**`stat -c %A file`**  
Display file permissions

**`stat -c %s file`**  
Display file size, in bytes

**`od file`**  
Dump a file into octal (or other formats)

**`hexdump options file`**  
Dump a file into hexadecimal (or octal, decimal, ASCII)

**`xxd options file`**  
Convert a file from binary to hexadecimal (i.e create a hex dump), or vice versa

**`uuencode file`**  
Encode a file using only printing ASCII characters. Used to transmit a binary file over a medium that supports only ASCII data, e.g. e-mail

**`uuencode -m file`**  
Encode a file to Base64

**`uudecode file`**  
Decode a file
In Linux, everything is (displayed as) a file. File descriptors are automatically associated to any process launched.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Default device</th>
<th>Device file</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Standard input (stdin)</td>
<td>Input text stream</td>
<td>Keyboard</td>
<td>/dev/stdin</td>
</tr>
<tr>
<td>1</td>
<td>Standard output (stdout)</td>
<td>Output text stream</td>
<td>Terminal</td>
<td>/dev/stdout</td>
</tr>
<tr>
<td>2</td>
<td>Standard error (stderr)</td>
<td>Output text stream</td>
<td>Terminal</td>
<td>/dev/stderr</td>
</tr>
</tbody>
</table>

```bash
mail user@email < file
```
Redirect file to the stdin of command mail (in this case, send via e-mail the contents of file to the email address user@email). Redirection is handled by the shell, not by the command invoked. The space after the redirection operator is optional.

```bash
ls > file
ls 1> file
ls >| file
ls >> file
ls 1>> file
```
Redirect the stdout of command ls to file (in this case, write on file the contents of the current directory). This overwrites file if it already exists, unless the Bash noclobber option is set (via set -o noclobber). Redirect the stdout of command ls to file, even if noclobber is set. Append the stdout of command ls to file.

```bash
ls 2> file
ls 2>> file
ls 2>/dev/null
```
Redirect the stderr of command ls to file (in this case, write any error encountered by the command ls to file). Append the stderr of command ls to file. Silence any error coming from the command ls.

```bash
cat <file1 >file2
<file1 cat >file2
<file1 >file2 cat
```
Redirect file1 to the stdin and file2 to the stdout of the command cat (in this case, copy file1 to file2). cat >file2 <file1 also works but is not recommended, because it truncates file2 if for some reason file1 cannot be opened.

```bash
cat /etc/passwd | wc -l
```
Pipe the stdout of command cat to the stdin of command wc (in this case, print the number of accounts in the system). Piped commands run concurrently.

```bash
echo "$(sort file)" > file
```
Sort the contents of file and write the output to the file itself. `sort file > file` would not produce the desired result, because the stdout destination is created (and therefore the content of the preexisting file is deleted) before the `sort` command is run.

```bash
ls 2>$1
ls > file 2>$1
```
Redirect stderr of command ls to stdout. Redirect both stdout and stderr of command ls to file. `ls > file` and `ls >$ file` also work on some systems but are not recommended, because they are not POSIX standard.

```bash
> file
```
Create an empty file. If the file exists, its content will be deleted.

```bash
ls | tee file
```
`tee` reads from stdin and writes both to stdout and file (in this case, writes the contents of the current directory to screen and to file at the same time).

```bash
ls | tee -a file
```
`tee` reads from stdin and appends both to stdout and file.
read MYVAR
read -n 8 MYVAR
read -t 60 MYVAR
read -s MYVAR

while read -r line
do
echo "Hello $line"
done < file

while read line
do
  for word in $line
do
    echo "Hello $word"
done
done < file

while IFS=$'\t' read -r -a array
do
  echo "${array[0]}"
  echo "${array[1]}"
  echo "${array[2]}"
done < file

echo $MYVAR

while read line
do
  echo "$line"
done < file

While read -r line
do
  echo "$line"
done < file

while read line
do
  for word in $line
do
    echo "$word"
done
done < file

while read line
do
  echo "$line"
done < file

while read -r line
do
  echo "$line"
done < file

while read line
do
  echo "$line"
done < file

while read line
do
  echo "$line"
done < file
Any application, program, script, or service that runs on the system is a **process**. Processes whose parent is a shell are called **jobs**.  
**Signals** are used for inter-process communication. Each process has a unique PID (Process ID) and a PPID (Parent Process ID); when a process spawns a child, the process PID is assigned to the child's PPID.  
The `/sbin/init` process, run at bootup, has PID 1. It is the ancestor of all processes and becomes the parent of any orphaned process. It is also unkillable; should it die, the kernel will panic.  
When a child process dies, its status becomes EXIT_ZOMBIE and a SIGCHLD is sent to the parent. The parent should then call the `wait()` system call to read the dead process' exit status and other information; until that moment, the child process remains a zombie.

ps -ef *(UNIX options)*
ps aux *(BSD options)*
ps tree PID

Display all processes in hierarchical format.

The process tree is rooted at PID, or at `init` if PID is omitted

pidof processname
pidof -s processname

Show PIDs of processes with name `processname`

Show PID of process with name `processname`, returning a single result

pgrep sshd
ps -ef | grep "[s]shd"
pgrep -u root sshd

Show processes whose name is "sshd"

Show processes whose name is "sshd" and are owned by root

pmap PID

Display the memory map of process `PID`

jobs

List all jobs

Suspend a job, putting it in the stopped state (send a SIGTSTP)

Put job # n in the background (send a SIGCONT)

Resume job # n in the foreground and make it the current job (send a SIGCONT)

Kill job # n

Remove job #n from the table of active jobs

Prevent job #n from receiving a SIGHUP if the shell receives that signal

Fork bomb: starts a process that continually replicates itself, slowing down or crashing the system because of resource starvation. Dangerous!

Run `command` and kill it after `n` seconds

To each process is associated a niceness value: the higher the niceness, the lower the priority. 
The niceness value ranges from -20 to 19, and a newly created process has a default niceness of 0. Unprivileged users can modify a process' niceness only within the range from 1 to 19.

nice -n -5 command

Start `command` with a niceness of -5. If niceness is omitted, a default value of 10 is used

renice -5 command

Change the niceness of a running `command` to -5

snice

Change the niceness of a process. Obsolete
<table>
<thead>
<tr>
<th>Signal number</th>
<th>Signal name</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIGHUP</td>
<td>Used by many daemons to reload their configuration</td>
</tr>
<tr>
<td>2</td>
<td>SIGINT</td>
<td>Interrupt, stop</td>
</tr>
<tr>
<td>9</td>
<td>SIGKILL</td>
<td>Kill unconditionally (this signal cannot be ignored)</td>
</tr>
<tr>
<td>15</td>
<td>SIGTERM</td>
<td>Terminate gracefully</td>
</tr>
<tr>
<td>18</td>
<td>SIGCONT</td>
<td>Continue execution</td>
</tr>
<tr>
<td>20</td>
<td>SIGTSTP</td>
<td>Stop execution</td>
</tr>
</tbody>
</table>

The manpage `man 7 signal` lists all signal numbers and names.

```
kill -l
List all available signal names
kill -l n
Print the name of signal number n

kill -9 1138
Send a signal 9 (SIGKILL) to process 1138, hence killing it
killall -9 sshd
Kill processes whose name is "sshd"
pkill -9 -u root sshd
Kill processes whose name is "sshd" and are owned by root
pkill -9 -u user
Kill all processes owned by user, forcing the user to log out
skill
Send a signal to a process or show process status. Obsolete
xkill
Kill a process by its X GUI resource. Pops up a cursor to select a window

nohup script.sh
Prevent a process from terminating (receiving a SIGHUP) when its parent Bash dies. When a Bash shell is terminated cleanly via `exit`, its jobs will become child of the Bash's parent and will continue running. When a Bash shell is killed instead, it issues a SIGHUP to its children which will terminate

trap action condition
Trap a signal
strace command
Trace the execution of `command`, intercepting and printing system calls called by a process and signals received by a process
ipcs
Show IPC facilities information (shared memory, message queues, and semaphores)
Resource monitoring

- **top**
  - Monitor processes in real-time
- **htop**
  - Monitor processes in real-time (ncurses UI)
- **iotop**
  - Display I/O usage by processes in the system
- **atop**
  - Advanced system monitor that displays the load on CPU, RAM, disk, and network
- **powertop**
  - Power consumption and power management diagnosis tool
- **uptime**
  - Show how long the system has been up, how many users are connected, and the system load averages for the past 1, 5, and 15 minutes
- **time command**
  - Execute `command` and, at its completion, write to stderr timing statistics about the run: elapsed real time between invocation and termination, user CPU time, system CPU time
- **sar**
  - Show reports about system activity (including reboots). Reports are generated from data collected via the cron job `sysstat` and stored in `/var/log/sa/sa{n}`, where `n` is the day of the month
  - `sar -f /var/log/sa/sa13 -s 06:00:00 -e 09:00:00` Show reports for system activity from 6 to 9 AM on the 13th of the month
  - `sar -u n m` Show real-time CPU activity, every `n` seconds for `m` times
  - `sar -n DEV` Show real-time network activity (received and transmitted packets per second)
- **sysbench**
  - Multi-threaded benchmark tool able to monitor different OS parameters: file I/O, scheduler, memory allocation, thread implementation, databases
- **inxi**
  - Debugging tool to rapidly and easily gather system information and configuration
- **stress-ng**
  - Tool for CPU and RAM stress tests
- **collectd**
  - System statistics collector

<table>
<thead>
<tr>
<th>Linux monitoring tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nagios</td>
</tr>
<tr>
<td>MRTG</td>
</tr>
<tr>
<td>Cacti</td>
</tr>
<tr>
<td>Munin</td>
</tr>
<tr>
<td>Zabbix</td>
</tr>
<tr>
<td>Centreon</td>
</tr>
<tr>
<td>Netdata</td>
</tr>
</tbody>
</table>
vmstat
Print a report about virtual memory statistics: processes, memory, paging, block I/O, traps, disks, and CPU activity

iostat
Print a report about CPU utilization, device utilization, and network filesystem. The first report shows statistics since the system boot; subsequent reports will show statistics since the previous report

mpstat
Print a report about processor activities

vmstat n m
iostat n m
mpstat n m
Print the relevant report every n seconds for m times

### Output of command vmstat

<table>
<thead>
<tr>
<th>procs</th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of runnable processes (running or waiting for run time)</td>
<td>Number of processes in uninterruptible sleep</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>memory</th>
<th>swpd</th>
<th>free</th>
<th>buff</th>
<th>cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual memory used (swap)</td>
<td>Free memory (idle)</td>
<td>Memory used as buffers</td>
<td>Memory used as cache</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>swap</th>
<th>si</th>
<th>so</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory swapped in from disk</td>
<td>Memory swapped out to disk</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>io</th>
<th>bi</th>
<th>bo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocks received in from a block device</td>
<td>Blocks sent out to a block device</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>system</th>
<th>in</th>
<th>cs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of interrupts</td>
<td>Number of context switches</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cpu</th>
<th>us</th>
<th>sy</th>
<th>id</th>
<th>wa</th>
<th>st</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time spent running user code (non-kernel)</td>
<td>Time spent running system code (kernel)</td>
<td>Time spent idle</td>
<td>Time spent waiting for I/O</td>
<td>Time stolen from a virtual machine</td>
<td></td>
</tr>
</tbody>
</table>

© Daniele Raffo dr0.ch
Show the amount of free and used memory in the system

<table>
<thead>
<tr>
<th>Output of command <code>free</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>total</strong> used free shared buff/cache available</td>
</tr>
<tr>
<td>Mem: 16344088 2273312 11531400 776228 2539376 12935112</td>
</tr>
<tr>
<td>Swap: 1048572 0 1048572</td>
</tr>
</tbody>
</table>

| **total** used free shared buffers cached |
| Mem: 1504544 1491098 13021 0 91112 764542 |
| Swap: 2047686 7667 2040019 |

Mem

- **total**: Total configured amount of memory
- **used**: Used memory
- **free**: Unused memory
- **shared**: Memory used by tmpfs, 0 if not available
- **buff/cache**: Memory used by kernel buffers, page cache, and slabs
- **available**: Memory available for new applications (without using swap)

-/+ buffers/cache

- **used**: Memory used by kernel buffers
- **free**: Memory available for new applications (without using swap)

Swap

- **total**: Total configured amount of swap space
- **used**: Used swap space
- **free**: Free swap space

* These are the true values indicating the free system resources available.

All values are in Kb, unless options are used.
### File permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Octal value</th>
<th>Command</th>
<th>Effect on file</th>
<th>Effect on directory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user: 400</td>
<td>chmod u+r</td>
<td>Can open and read the file</td>
<td>Can list directory content</td>
<td></td>
</tr>
<tr>
<td>group: 40</td>
<td>chmod g+r</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>others: 4</td>
<td>chmod o+r</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Write</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user: 200</td>
<td>chmod u+w</td>
<td>Can modify the file</td>
<td>Can create, delete, and rename files in the directory</td>
<td></td>
</tr>
<tr>
<td>group: 20</td>
<td>chmod g+w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>others: 2</td>
<td>chmod o+w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Execute</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user: 100</td>
<td>chmod u+x</td>
<td>Can execute the file (binary or script)</td>
<td>Can enter the directory, and search files within (by accessing a file's inode)</td>
<td></td>
</tr>
<tr>
<td>group: 10</td>
<td>chmod g+x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>others: 1</td>
<td>chmod o+x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SetUID (SUID)</strong></td>
<td>4000</td>
<td>chmod u+s</td>
<td>Executable is run with the privileges of the file’s owner</td>
<td>No effect</td>
</tr>
<tr>
<td><strong>SetGID (SGID)</strong></td>
<td>2000</td>
<td>chmod g+s</td>
<td>Executable is run with the privileges of the file’s group</td>
<td>All new files and subdirectories inherit the directory’s group ID</td>
</tr>
<tr>
<td><strong>Sticky</strong></td>
<td>1000</td>
<td>chmod +t</td>
<td>No effect</td>
<td>Files inside the directory can be deleted or moved only by the file’s owner</td>
</tr>
</tbody>
</table>

- `chmod 711 file` Set read, write, and execute permission to user; set execute permission to group and others
- `chmod u=rwx,go=x file` Add read and execute permission to user
- `chmod -x file` Remove execute permission from everybody (user, group, and others)
- `chmod -R g+x /path` Set the group execute bit recursively on path and every dir and file underneath
- `find /path -type d -exec chmod g+x {} \;` Set the group execute bit recursively on path and every dir, but not file, underneath
- `chown user file` Change the owner of the file to user
- `chown user:group file` Change the owner of the file to user, and group ownership of the file to group
- `chgrp group file` Change group ownership of the file to group
- `umask 022` Set the permission mask to 022, hence masking write permission for group and others.

Linux default permissions are 0666 for files and 0777 for directories. These base permissions are ANDed with the inverted umask value to calculate the final permissions of a new file or directory.
File attributes

- [ ] = regular file
  - = directory
  l = symbolic link
  s = Unix domain socket
  p = named pipe
  c = character device file
  b = block device file

. = file with SELinux context
+ = file with ACL

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>File can only be opened in append mode for writing</td>
</tr>
<tr>
<td>A</td>
<td>When file is accessed, its atime record is not modified. This reduces I/O operations</td>
</tr>
<tr>
<td>c</td>
<td>File is automatically compressed on the fly on disk by the kernel. Reading it produces uncompressed data</td>
</tr>
<tr>
<td>C</td>
<td>For filesystems which perform copy-on-write, file is not subject to copy-on-write updates</td>
</tr>
<tr>
<td>d</td>
<td>File will not be backed up by the dump program</td>
</tr>
<tr>
<td>D</td>
<td>When directory is modified, changes are written synchronously on disk. Equivalent to the dirsync mount option</td>
</tr>
<tr>
<td>e</td>
<td>File is using extents for mapping the blocks on disk</td>
</tr>
<tr>
<td>E</td>
<td>Compression error on a compressed file. This attribute is used by experimental compression patches</td>
</tr>
<tr>
<td>h</td>
<td>File stores its blocks in units of filesystem blocksize instead of in units of sectors. This means that the file is (or was) larger than 2 Tb</td>
</tr>
<tr>
<td>i</td>
<td>File is immutable i.e. cannot be deleted, modified, renamed, linked, or changed permissions</td>
</tr>
<tr>
<td>I</td>
<td>Directory is being indexed using hashed trees</td>
</tr>
<tr>
<td>j</td>
<td>All file data is written to the ext3 or ext4 journal before being written to the file itself</td>
</tr>
<tr>
<td>N</td>
<td>File has data stored inline within the inode itself</td>
</tr>
<tr>
<td>s</td>
<td>File will be securely wiped by zeroing when deleted</td>
</tr>
<tr>
<td>S</td>
<td>When file is modified, changes are written synchronously on disk. Equivalent to the sync mount option</td>
</tr>
<tr>
<td>t</td>
<td>For filesystems with support for tail-merging, file will not have EOF partial block fragment merged with other files. This is necessary for some applications e.g. LILO</td>
</tr>
<tr>
<td>T</td>
<td>Directory is the top of directory hierarchies for the purpose of the Orlov block allocator</td>
</tr>
<tr>
<td>u</td>
<td>File can be undeleted after being deleted</td>
</tr>
<tr>
<td>X</td>
<td>Raw contents of compressed file can be accessed directly. This attribute is used by experimental compression patches</td>
</tr>
<tr>
<td>Z</td>
<td>Compressed file is dirty. This attribute is used by experimental compression patches</td>
</tr>
</tbody>
</table>

`chattr +attribute file` | Add a file or directory attribute |
`chattr -attribute file` | Remove a file or directory attribute |
`chattr =attribute file` | Set a file or directory attribute, removing all other attributes |
`lsattr file` | List file or directory attributes |

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Value tracked</th>
<th>Displayed via</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtime</td>
<td>Time of last modification to file contents (data itself)</td>
<td><code>ls -l</code></td>
</tr>
<tr>
<td>ctime</td>
<td>Time of last change to file contents or file metadata (owner, group, or permissions)</td>
<td><code>ls -lc</code></td>
</tr>
<tr>
<td>atime</td>
<td>Time of last access to file for reading contents</td>
<td><code>ls -lu</code></td>
</tr>
</tbody>
</table>

The POSIX standard does not define a timestamp for file creation. Some filesystems (e.g. ext4, JFS, Btrfs) store this value, but currently there is no Linux kernel API to access it.
Access Control Lists (ACLs) provide a fine-grained set of permissions that can be applied to files and directories. An access ACL is set on an individual file or directory; a default ACL is set on a directory, and applies to all files and subdirs created inside it that don't have an access ACL. The final permissions are the intersection of the ACL with the chmod/umask value. A partition must have been mounted with the acl option in order to support ACLs on files.

setfacl -m u: user:permissions file
Set an access ACL on a file for a user

setfacl -m g: group:permissions file
Set an access ACL on a file for a group

setfacl -m m: permissions file
Set the effective rights mask on a file

setfacl -m o: permissions file
Remove an access ACL from a file for other users

setfacl -x u: user file
Remove an access ACL from a file for a user

The permissions are standard Unix permissions specified as any combination of rwx.

setfacl -m d: u: user:permissions dir
Same as above, but set a default ACL instead of an access ACL.
This applies to all commands above

setfacl -d -m u: user:permissions dir

getfacl file
Display the access (and default, if any) ACL for a file

getfacl file1 | setfacl --set-file= file2
Copy the ACL of file1 and apply it to file2

getfacl --access dir | setfacl -d -M dir
Copy the access ACL of a directory and set it as default ACL

chacl options
Change an ACL.
This command exists to provide compatibility with IRIX

man acl
Show the manpage about ACLs
An **inode** is a structure containing all file metadata: file type, permissions, owner, group, size, access/change/modification/deletion times, number of links, attributes, ACLs, and address where the actual file content (data) is stored. However, an inode does not contain the name of the file; this information is stored in the directory where the file is located (i.e. referenced).

A directory contains a list of mappings between filenames and inodes.

In Linux, two types of links exist: **hard links** and **symbolic links** (aka **soft links**).

The **link count** of a file is the total number of hard links to that file (i.e. to that file’s inode). By default, files have a link count of 1, and directories have a link count of 2 (the directory itself, and the . link inside the directory). The link count of a directory is increased by one for each subdirectory (because of the .. parent link inside the subdirectory). Once a file has no hard links pointing to it, the file is deleted, provided that no process holds the file open for reading.

<table>
<thead>
<tr>
<th><strong>Hard link</strong></th>
<th><strong>Symbolic link</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition</strong></td>
<td>A link to an already existing inode</td>
</tr>
<tr>
<td><strong>Command to create it</strong></td>
<td><code>ln file hardlink</code></td>
</tr>
<tr>
<td><strong>Link is still valid if the original file is moved or deleted</strong></td>
<td>Yes (because the link still references the inode to which the original file pointed)</td>
</tr>
<tr>
<td><strong>Can link to a file in another filesystem</strong></td>
<td>No (because inode numbers make sense only within a determinate filesystem)</td>
</tr>
<tr>
<td><strong>Can link to a directory</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Link permissions</strong></td>
<td>Reflect the original file’s permissions, even when these are changed</td>
</tr>
<tr>
<td><strong>Link attributes</strong></td>
<td><code>-</code> (regular file)</td>
</tr>
<tr>
<td><strong>Inode number</strong></td>
<td>The same as the original file</td>
</tr>
</tbody>
</table>

- `ls -i` Show a listing of the directory with the inode number for each file
- `ls -l` Show a listing of the directory with the link count for each file
- `df -i` Report filesystem inode usage
- `find / -inum n` Find all files linked to the same inode n
- `find / -samefile file` Find all files linked to the same inode as file
Find system files

```sh
cat
```

find /path -name "foo*"
find / -name "foo*" -exec chmod 700 {} \;
find / -name "foo*" -ok chmod 700 {} \;
find / -size +128M
find / -type f -ctime +10
find / -type f -perm -4000
find / -type f -newermt "May 4 2:55" -delete
find . -type f -print -exec cat {} \;
find . \! -name "*.gz" -type f -exec gzip {} \;
find / -xdev -type f -size +100M -exec ls -lah {} \;
```

locate file
slocate file
updatedb

which command
which -a command

whereis command
whereis -b command
whereis -s command
whereis -m command

type command

file file

Find all files and dirs, in the directory tree rooted at /path, whose name starts with "foo"
Find all files and dirs whose name start with "foo" and apply permission 700 to all of them
Find all files and dirs whose name start with "foo" and apply permission 700 to all of them, asking for confirmation
Find all files larger than 128 Mb
Find all files last changed more than 10 days ago
Find all files with SUID set (a possible security risk, because a shell with SUID root is a backdoor)
Find and delete all files newer than the specified timestamp. Using -delete is preferable to using -exec rm {} \;
Print all files, in the current directory and under, precluding them with a filename header
Find all files, in the current directory and under, which do not have the gz extension, and compress them
Find all files larger than 100 Mb in the current filesystem only and display detailed information about them

Locate file by searching the file index /etc/updatedb.conf, not by actually walking the filesystem. The search is fast but will only held results relative to the last rebuild of the file index
Rebuild the file index

Locate a binary executable command within the PATH
Locate all matches of a command, not only the first one
Locate the binary, source, and manpage files for a command
Locate the binary files for a command
Locate the source files for a command
Locate the manpage files for a command

Determine if a command is a program or a built-in (i.e. an internal feature of the shell)

Analyze the content of a file or directory, and display the kind of file (e.g. executable, text file, program text, swap file)
Shell usage

Show the history of command lines executed up to this moment. Commands prepended by a space will be executed but will not show up in the history. After the user logs out from Bash, history is saved into ~/.bash_history

!n

Execute command number n in the command line history

history -c
Clear the command line history

history -d n
Delete command number n from the command line history

export HISTSIZE=m
Set the command line history to contain only the m past commands

alias ls='ls -lap'
Set up an alias for the ls command

alias
Show defined aliases

unalias ls
Remove the alias for the ls command

ls
Run the non-aliased version of the ls command

/bin/ls

Almost all Linux commands accept the option -v (verbose), and some commands also accept the options -vv or -vvv (increasing levels of verbosity).

All Bash built-in commands, and all commands respecting the POSIX requirements, accept the flag -- which denotes the end of options and the start of positional parameters:

grep -- -i file
Search for the string "-i" in file

rm -- -rf
Delete a file called "-rf"

Another way to operate on a file whose name begins with a special character is to specify the path:

rm ./-rf
Delete a file called "-rf"

cat /etc/debian_version
Display Linux distribution name and version

(Debian)

(Same as)

cat /etc/fedora-release
Display Linux distribution name and version

(Fedora)

cat /etc/redhat-release
Display Linux distribution name and version

(Red Hat)

cat /etc/lsb-release
Display Linux distribution name and version

lsb_release -a

cat /etc/os-release
Display Linux distribution name and version

Display Linux distribution name and version

Shells

<table>
<thead>
<tr>
<th>Shell</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sh</td>
<td>Thompson Shell</td>
</tr>
<tr>
<td>sh</td>
<td>Bourne Shell</td>
</tr>
<tr>
<td>bash</td>
<td>Bash (Bourne Again Shell)</td>
</tr>
<tr>
<td>csh</td>
<td>C shell</td>
</tr>
<tr>
<td>tcsh</td>
<td>tcsh</td>
</tr>
<tr>
<td>ksh</td>
<td>KornShell</td>
</tr>
<tr>
<td>zsh</td>
<td>Z shell</td>
</tr>
<tr>
<td>ash</td>
<td>Almquist shell</td>
</tr>
<tr>
<td>dash</td>
<td>Debian Almquist shell</td>
</tr>
<tr>
<td>fish</td>
<td>Friendly interactive shell</td>
</tr>
</tbody>
</table>
Shell variables

The scope of variables is the current shell only, while environment variables are visible within the current shell as well as within all subshells and Bash child processes spawned by the shell. Environment variables are set in /etc/environment in the form `variable=value`. Conventionally, variable names are lowercase while environment variable names are uppercase.

- `set` Display all variables
- `env` Display all environment variables
- `readonly -p` Display all variables that are read-only

```bash
VAR=value
((VAR=value))
let "VAR=value"
readonly VAR=value
set ${VAR=value}
VAR=${VAR:-value}
unset VAR
export VAR
```

- `command $VAR` Pass a variable as argument to `command`
- `command ${VAR} HELLO` If other characters follow the variable name, it is necessary to specify the boundaries of the variable name via `()` to make it unambiguous.
- `command "${VAR}"` It is recommended to double quote the variable when referencing it, to prevent interpretation of special characters (except `\` and ` `) and word splitting (in case the variable value contains whitespaces), both of which will cause unintended results

```bash
VAR=$((5 + 37))
VAR=$[5 + 37]
VAR=$((VAR2 + 42))
VAR='expr $VAR2 + 42'
```

- `((VAR++))` Increase a variable by 1
- `((++VAR))` (variable value contains whitespaces), both of which will cause unintended results

```bash
VAR=`command`
VAR=$(command)
```

- `for i in /path/* do
echo "Filename: $i"
done` Loop and operate through all the output tokens (in this case, files in the `path`). The construct `for i in $(ls /path/)` must not be used, because filenames containing particular characters (whitespaces, glob characters, hyphens etc.) will cause unintended results

```bash
echo ${VAR:-message}
echo ${VAR:+message}
echo ${VAR,,}
```

- `echo $({STRING})` String tokenizer. Splits a string stored in the variable `STRING` into tokens, according to the content of the shell variable `$IFS`, and stores them in the array `TOKENS`

```bash
TOKENS=(${STRING})
```

```bash
echo ${TOKENS[n]}
echo ${TOKENS[*]}
```
**Shell mechanics**

### Bash built-in variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$0</code></td>
<td>Script name</td>
</tr>
<tr>
<td><code>$n</code></td>
<td>nth argument passed to the script or function</td>
</tr>
<tr>
<td><code>$@</code></td>
<td>All arguments passed to the script or function; each argument is a separate word</td>
</tr>
<tr>
<td><code>$*</code></td>
<td>All arguments passed to the script or function, as a single word</td>
</tr>
<tr>
<td><code>$#</code></td>
<td>Number of arguments passed to the script or function</td>
</tr>
<tr>
<td><code>$?</code></td>
<td>Exit status of the last recently executed command</td>
</tr>
<tr>
<td><code>${PIPESTATUS[n]}</code></td>
<td>Exit status of the nth command in the executed pipeline</td>
</tr>
<tr>
<td><code>$$</code></td>
<td>PID of the script in which this variable is called</td>
</tr>
<tr>
<td><code>$!</code></td>
<td>PID of the last recently executed background command</td>
</tr>
<tr>
<td><code>$SHLVL</code></td>
<td>Deepness level of current shell, starting with 1</td>
</tr>
<tr>
<td><code>$IFS</code></td>
<td>Internal Field Separator; defines what are the token separators for strings (e.g. for word splitting after expansion). By default it has the value &quot;space, tab, newline&quot;</td>
</tr>
<tr>
<td><code>$RANDOM</code></td>
<td>Pseudorandom integer value between 0 and 32767</td>
</tr>
</tbody>
</table>

### Bash shell event | Files run

<table>
<thead>
<tr>
<th>Event</th>
<th>Files run</th>
</tr>
</thead>
<tbody>
<tr>
<td>When a login shell is launched</td>
<td><code>/etc/profile</code></td>
</tr>
<tr>
<td></td>
<td><code>/etc/profile.d/*.sh</code></td>
</tr>
<tr>
<td></td>
<td><code>~/.bash_profile</code></td>
</tr>
<tr>
<td></td>
<td><code>~/.bash_login</code></td>
</tr>
<tr>
<td></td>
<td><code>~/.profile</code></td>
</tr>
<tr>
<td>When a login shell exits</td>
<td><code>~/.bash_logout</code></td>
</tr>
<tr>
<td>When a non-login shell is launched</td>
<td><code>/etc/bash.bashrc</code></td>
</tr>
<tr>
<td></td>
<td><code>/etc/bashrc</code></td>
</tr>
<tr>
<td></td>
<td><code>~/.bashrc</code></td>
</tr>
</tbody>
</table>
Shell options

- `set -option` Enable a Bash option
- `set -o longoption` Show the status of all Bash options
- `set +option` Disable a Bash option
- `set +o longoption` Print shell input lines as they are read
- `set -o verbose` Print command traces before execution of each command (debug mode)
- `set -e` Exit the script immediately if a command fails. Recommended option
- `set -u` Treat expansion of unset variables as an error. This avoids unintended results

There are three ways to run a script with a specific Bash option enabled:
- Run the script with `bash -option script.sh`
- Specify the shebang line in the script as `#!/bin/bash -option`
- Add the command `set -option` at the beginning of the script

- `shopt` Display the list of all shell options with their current value (on or off)
- `shopt -s shelloption` Set (enable) a specific shell option
- `shopt -u shelloption` Unset (disable) a specific shell option
Bash shell scripts must start with the shebang line `#!/bin/bash` indicating the location of the script interpreter.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>source script.sh</code></td>
<td>Script execution takes place in the same shell. Variables defined and exported in the script are seen by the shell when the script exits.</td>
</tr>
<tr>
<td><code>bash script.sh</code></td>
<td>Script execution spawns a new shell</td>
</tr>
</tbody>
</table>

- `command &` Execute `command` in the background
- `command1; command2` Execute `command 1` and then `command 2`
- `command1 && command2` Execute `command 2` only if `command 1` executed successfully (exit status = 0)
- `command1 || command2` Execute `command 2` only if `command 1` did not execute successfully (exit status > 0)
- `(command1 && command2)` Group commands together for evaluation priority
- `command || exit 1` Run `command` in a subshell. This is used to isolate `command`’s effects, as variable assignments and other changes to the shell environment operated by `command` will not remain after `command` completes
- `exit` Terminate a script
- `exit n` Terminate a script with the specified exit status number `n`. By convention, a 0 exit status is used if the script executed successfully, a non-zero value otherwise (To be used inside a script.) Exit the script if `command` fails
- `/bin/true` Do nothing and return immediately a status code of 0 (indicating success)
- `/bin/false` Do nothing and return immediately a status code of 1 (indicating failure)
- `if command then echo "Success" else echo "Failure" fi` Run a command, then evaluate whether it exited successfully or failed
- `function myfunc { commands }` Define a function. A function must be defined before it can be used in a Bash script. Argument number `n` is accessed in the body of the function via `$n`.
- `readonly -f myfunc` An advantage of functions over aliases is that functions can be passed arguments
- `readonly -p -f` Call a function
- `typeset -f` Mark an already defined function as read-only, preventing it from being redefined
- `myfunc arg1 arg2 ...` Show functions defined in the current Bash session
- `readonly -f` Show functions which are read-only
- `zenity` Display GTK+ graphical dialogs for user messages and input
getopts

Parse positional parameters in a shell script

```
getopts syntax

while getopts abc:d: OPT do
  case $OPT in
    a)
      command_a
      exit 0
    ;;
    b)
      command_b
      exit 0
    ;;
    c)
      command_c $OPTARG
      exit 0
    ;;
    d)
      command_d $OPTARG
      exit 0
    ;;
    *)
      default_command
      exit 1
    ;;
  esac
done

Definition of accepted options

Matches option -a.
Executes a command

Matches option -c argument.
Executes a command with argument

Command to execute if none of above options applies
```
watch command

Execute command every 2 seconds

watch -d -n 1 command

Execute command every second, highlighting the differences in the output

timeout 30s command

Execute command and kill it after 30 seconds

command | ts

Prepend a timestamp to each line of the output of command

sleep 5

Pause for 5 seconds

sleep ${($RANDOM % 60) + 1}s

Sleep for a random time between 1 and 60 seconds

sleep infinity

Pause forever

usleep 5000

Pause for 5000 microseconds

yes

Output endlessly the string "y"

yes string

Output endlessly string

yes | fsck /dev/sda

Automatically answer yes every time fsck asks for confirmation before fixing errors

script file

Generate a typescript of a terminal session. Forks a subshell and starts recording on file everything that is printed on terminal; the typescript ends when the user exits the subshell

expect

Dialogue with interactive programs according to a script, analyzing what can be expected from the interactive program and replying accordingly

cmdtest

Tool for black box testing of Linux command line programs

busybox

BusyBox, "the Swiss Army knife of Embedded Linux", an optimized multi-call binary which contains many Linux commands and utilities. Useful for system recovery if Bash built-ins or common commands have become unusable or have been removed from the system

busybox applet arguments

Execute applet, which operates as the homonym Linux command

xargs command

Call command multiple times, one for each argument found on stdin

ls foo* | xargs cat

Print via cat the content of every file whose name starts by "foo"

parallel command

Run command in parallel. This is used to operate on multiple inputs, similarly to xargs
test "$MYVAR" operator "value" && command
[ "$MYVAR" operator "value" ] && command
if [ "$MYVAR" operator "value" ]; then command; fi

Perform a test; if it results true, command is executed

<table>
<thead>
<tr>
<th>Test operators</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Integer operators</strong></td>
</tr>
<tr>
<td>-eq value</td>
</tr>
<tr>
<td>-ne value</td>
</tr>
<tr>
<td>-lt value</td>
</tr>
<tr>
<td>-le value</td>
</tr>
<tr>
<td>-gt value</td>
</tr>
<tr>
<td>-ge value</td>
</tr>
<tr>
<td><strong>Numeric operators</strong></td>
</tr>
<tr>
<td>= value</td>
</tr>
<tr>
<td>!= value</td>
</tr>
<tr>
<td>&lt; value</td>
</tr>
<tr>
<td>&lt;= value</td>
</tr>
<tr>
<td>&gt; value</td>
</tr>
<tr>
<td>&gt;= value</td>
</tr>
<tr>
<td><strong>Expression operators</strong></td>
</tr>
<tr>
<td>expr1 -a expr2</td>
</tr>
<tr>
<td>expr1 -o expr2</td>
</tr>
<tr>
<td>! expr</td>
</tr>
<tr>
<td>( expr )</td>
</tr>
<tr>
<td><strong>String operators</strong></td>
</tr>
<tr>
<td>-z</td>
</tr>
<tr>
<td>-n or nothing</td>
</tr>
<tr>
<td>= string or == string</td>
</tr>
<tr>
<td>!= string</td>
</tr>
<tr>
<td>&lt; string</td>
</tr>
<tr>
<td>&gt; string</td>
</tr>
<tr>
<td>substr string pos len</td>
</tr>
<tr>
<td>length string</td>
</tr>
<tr>
<td>string : regex</td>
</tr>
<tr>
<td>or match string regex</td>
</tr>
</tbody>
</table>

expr "$MYVAR" = "39 + 3"  Evaluate an expression (in this case, assigns the value 42 to the variable)
expr string : regex  Return the length of the substring matching the regex
expr string : \(\text{regex}\)  Return the substring matching the regex
## Operators

<table>
<thead>
<tr>
<th>Mathematical operators</th>
<th>Logical operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Addition</td>
<td>!</td>
</tr>
<tr>
<td>-</td>
<td>&amp;&amp;</td>
</tr>
<tr>
<td>Subtraction</td>
<td>Logical negation</td>
</tr>
<tr>
<td>*</td>
<td>Logical AND</td>
</tr>
<tr>
<td>Multiplication</td>
<td>Logical OR</td>
</tr>
<tr>
<td>/</td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td></td>
</tr>
<tr>
<td>Remainder</td>
<td></td>
</tr>
<tr>
<td>**</td>
<td></td>
</tr>
<tr>
<td>Exponentiation</td>
<td></td>
</tr>
<tr>
<td>++</td>
<td></td>
</tr>
<tr>
<td>Pre/post increment</td>
<td></td>
</tr>
<tr>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Pre/post decrement</td>
<td></td>
</tr>
</tbody>
</table>

### Bitwise operators

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>~</td>
</tr>
<tr>
<td>Bitwise negation</td>
</tr>
<tr>
<td>&amp;</td>
</tr>
<tr>
<td>Bitwise AND</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Bitwise OR</td>
</tr>
<tr>
<td>^</td>
</tr>
<tr>
<td>Bitwise XOR</td>
</tr>
<tr>
<td>&lt;&lt;</td>
</tr>
<tr>
<td>Left bitwise shift</td>
</tr>
<tr>
<td>&gt;&gt;</td>
</tr>
<tr>
<td>Right bitwise shift</td>
</tr>
</tbody>
</table>

### Assignment operators

| =                  |
| Assignment         |
| op=                |
| Operation and assignment |
Flow control

Tests

```bash
if [ test 1 ]
    then
        [command block 1]
    elif [ test 2 ]
    then
        [command block 2]
    else
        [command block 3]
fi
```

```bash
case $STRING in
    pattern1)
        [command block 1]
    ;;
    pattern2)
        [command block 2]
    ;;
    *)
        [command block default]
    ;;
esac
```

Loops

```bash
while [test]
do
    [command block]
done
```

```bash
until [test]
do
    [command block]
done
```

```bash
for item in [list]
do
    [command block]
done
```

```bash
while [test]
do
    [command block]
done
```

```bash
until [test]
do
    [command block]
done
```

```bash
for i in 0 1 2 3 4 5 6 7
done
```

```bash
for i in {0..7}
done
```

```bash
for i in $(seq $start $end)
done
```

```bash
for ((i = start; i <= end; i++))
done
```

Loop breaking

```bash
break
```

```bash
Exit a loop
```

```bash
continue
```

```bash
Jump to the next iteration
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vi</td>
<td>Vi, text editor</td>
</tr>
<tr>
<td>vim</td>
<td>Vi Improved, an advanced text editor</td>
</tr>
<tr>
<td>gvim</td>
<td>Vim with GUI</td>
</tr>
<tr>
<td>vimdif f1 f2</td>
<td>Compare two text files in Vim</td>
</tr>
<tr>
<td>pico</td>
<td>Pico, simple text editor</td>
</tr>
<tr>
<td>nano</td>
<td>Nano, simple text editor (a GNU clone of Pico)</td>
</tr>
<tr>
<td>rnano</td>
<td>Restricted version of Nano: does not allow the user access the filesystem (except for files specified as argument) or a command shell</td>
</tr>
<tr>
<td>emacs</td>
<td>GNU Emacs, a GUI text editor</td>
</tr>
<tr>
<td>gedit</td>
<td>GUI text editor</td>
</tr>
<tr>
<td>ed</td>
<td>Line-oriented text editor</td>
</tr>
<tr>
<td>hexedit</td>
<td>Hexadecimal and ASCII editor</td>
</tr>
<tr>
<td>more</td>
<td>Text pager (obsolete)</td>
</tr>
<tr>
<td>less</td>
<td>Text pager</td>
</tr>
<tr>
<td>most</td>
<td>Text pager with advanced features (screen split, binary viewer, etc.)</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>G</code></td>
<td>Go to the first line in the file</td>
</tr>
<tr>
<td><code>n</code></td>
<td>Go to line number <code>n</code></td>
</tr>
<tr>
<td><code>G</code></td>
<td>Go to the last line in the file</td>
</tr>
<tr>
<td><code>F</code></td>
<td>Go to the end of the file, and move forward automatically as the file grows</td>
</tr>
<tr>
<td><code>CTRL-C</code></td>
<td>Stop moving forward</td>
</tr>
<tr>
<td><code>-N</code></td>
<td>Show line numbers</td>
</tr>
<tr>
<td><code>-n</code></td>
<td>Don't show line numbers</td>
</tr>
<tr>
<td><code>=</code></td>
<td>Show information about the file</td>
</tr>
<tr>
<td><code>CTRL-G</code></td>
<td>Show current and total line number, byte, and percentage of the file read</td>
</tr>
<tr>
<td><code>/pattern</code></td>
<td>Search <code>pattern</code> forward</td>
</tr>
<tr>
<td><code>?pattern</code></td>
<td>Search <code>pattern</code> backwards</td>
</tr>
<tr>
<td><code>&amp;pattern</code></td>
<td>Display only lines matching <code>pattern</code></td>
</tr>
<tr>
<td><code>n</code></td>
<td>Search next occurrences forward</td>
</tr>
<tr>
<td><code>N</code></td>
<td>Search next occurrences backwards</td>
</tr>
<tr>
<td><code>:n</code></td>
<td>When reading multiple files, go to the next file</td>
</tr>
<tr>
<td><code>:p</code></td>
<td>When reading multiple files, go to the previous file</td>
</tr>
<tr>
<td><code>R</code></td>
<td>Repaint the screen</td>
</tr>
<tr>
<td><code>V</code></td>
<td>Show version number</td>
</tr>
<tr>
<td><code>h</code></td>
<td>Help</td>
</tr>
<tr>
<td><code>q</code></td>
<td>Quit</td>
</tr>
</tbody>
</table>

**Command Line**

- `less +command file` Open file for reading, applying command (see list above)
- `less +F --follow-name file` Move forward, attempting periodically to reopen file by name; useful to keep reading a logfile that is being rotated. Note that, by default, `less` continues to read the original input file even if it has been renamed.
**ESC**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>Insert text before cursor and go to Insert mode</td>
</tr>
<tr>
<td>I</td>
<td>Insert text after line</td>
</tr>
<tr>
<td>a</td>
<td>Append text after cursor</td>
</tr>
<tr>
<td>A</td>
<td>Append text after line</td>
</tr>
<tr>
<td>v</td>
<td>Go to Visual mode, character-wise then use the arrow keys to select a block of text</td>
</tr>
<tr>
<td>V</td>
<td>Go to Visual mode, line-wise</td>
</tr>
<tr>
<td>d</td>
<td>Delete selected block</td>
</tr>
<tr>
<td>y</td>
<td>Copy (yank) selected block into buffer</td>
</tr>
<tr>
<td>w</td>
<td>Move to next word</td>
</tr>
<tr>
<td>b</td>
<td>Move to beginning of word</td>
</tr>
<tr>
<td>e</td>
<td>Move to end of word</td>
</tr>
<tr>
<td>o</td>
<td>Move to beginning of line</td>
</tr>
<tr>
<td>g</td>
<td>Move to end of file</td>
</tr>
<tr>
<td>1G</td>
<td>Move to line 1 i.e. beginning of file</td>
</tr>
<tr>
<td>G</td>
<td>Move to end of file</td>
</tr>
<tr>
<td>0</td>
<td>Move to beginning of line</td>
</tr>
<tr>
<td>z</td>
<td>Make current line the top line of the screen</td>
</tr>
<tr>
<td>ma</td>
<td>Mark position &quot;a&quot;. Marks a-z are local to current file, while marks A-Z are global to a specific file</td>
</tr>
<tr>
<td>'a</td>
<td>Go to mark &quot;a&quot;. If using a global mark, it also opens the specific file</td>
</tr>
<tr>
<td>y'a</td>
<td>Copy (yank) from mark &quot;a&quot; to current line, into the buffer</td>
</tr>
<tr>
<td>d'a</td>
<td>Delete from mark &quot;a&quot; to current line</td>
</tr>
<tr>
<td>p</td>
<td>Paste buffer after current line</td>
</tr>
<tr>
<td>P</td>
<td>Paste buffer before current line</td>
</tr>
<tr>
<td>x</td>
<td>Delete current character</td>
</tr>
<tr>
<td>X</td>
<td>Delete before current character</td>
</tr>
<tr>
<td>?dd</td>
<td>Delete 7 lines. Almost any command can be prepended by a number to repeat it that number of times</td>
</tr>
<tr>
<td>u</td>
<td>Undo last command. Vi can undo the last command only, Vim is able to undo several commands</td>
</tr>
<tr>
<td>-</td>
<td>Repeat last text-changing command</td>
</tr>
<tr>
<td>/string</td>
<td>Search for string forward</td>
</tr>
<tr>
<td>?string</td>
<td>Search for string backwards</td>
</tr>
<tr>
<td>:s/s1/s2/</td>
<td>Replace the first occurrence of s1 with s2 in the current line</td>
</tr>
<tr>
<td>:s/s1/s2/g</td>
<td>Replace globally every occurrence of s1 with s2 in the current line</td>
</tr>
<tr>
<td>:%s/s1/s2/g</td>
<td>Replace globally every occurrence of s1 with s2 in the whole file</td>
</tr>
<tr>
<td>:%s/s1/s2/gc</td>
<td>Replace globally every occurrence of s1 with s2 in the whole file, asking for confirmation</td>
</tr>
<tr>
<td>s5,40a/^#/</td>
<td>Add a hash character at the beginning of each line, from line 5 to 40</td>
</tr>
</tbody>
</table>

**CTRL + G**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:r file</td>
<td>Read file and insert it after current line</td>
</tr>
<tr>
<td>:X</td>
<td>Encrypt current document. Vi will automatically prompt for the password to encrypt and decrypt</td>
</tr>
<tr>
<td>:w file</td>
<td>Write to file</td>
</tr>
<tr>
<td>:wq</td>
<td>Save changes and quit</td>
</tr>
<tr>
<td>:x</td>
<td>Quit (fails if there are unsaved changes)</td>
</tr>
<tr>
<td>ZZ</td>
<td>Abandon all changes and quit</td>
</tr>
<tr>
<td>:q</td>
<td>Quit (fails if there are unsaved changes)</td>
</tr>
<tr>
<td>:q!</td>
<td>Abandon all changes and quit</td>
</tr>
</tbody>
</table>

**vi -R file**

Open file in read-only mode

**cat file | vi -**

Open file in read-only mode; this is done from the shell, by having Vi read from stdin
<table>
<thead>
<tr>
<th>Option</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>ai</td>
<td>Turn on auto indentation</td>
</tr>
<tr>
<td>all</td>
<td>Display all options</td>
</tr>
<tr>
<td>ap</td>
<td>Print a line after the commands</td>
</tr>
<tr>
<td>aw</td>
<td>Automatic write on commands :n ! e# ^w :rew ^j :tag</td>
</tr>
<tr>
<td>bf</td>
<td>Discard control characters from input</td>
</tr>
<tr>
<td>dir=tmpdir</td>
<td>Set tmpdir as directory for temporary files</td>
</tr>
<tr>
<td>eb</td>
<td>Precede error messages with a bell</td>
</tr>
<tr>
<td>ht=8</td>
<td>Set terminal tab as 8 spaces</td>
</tr>
<tr>
<td>ic</td>
<td>Ignore case when searching</td>
</tr>
<tr>
<td>lisp</td>
<td>Modify brackets for Lisp compatibility</td>
</tr>
<tr>
<td>list</td>
<td>Show tabs and EOL characters</td>
</tr>
<tr>
<td>set listchars=tab:&gt;-&gt;</td>
<td>Show tab as &gt; for the first char and as – for the following chars</td>
</tr>
<tr>
<td>magic</td>
<td>Allow pattern matching with special characters</td>
</tr>
<tr>
<td>mesg</td>
<td>Enable UNIX terminal messaging</td>
</tr>
<tr>
<td>nu</td>
<td>Show line numbers</td>
</tr>
<tr>
<td>opt</td>
<td>Speed up output by eliminating automatic Return</td>
</tr>
<tr>
<td>para=LI1PLPPPQbP</td>
<td>Set macro to start paragraphs for { } operators</td>
</tr>
<tr>
<td>prompt</td>
<td>Prompt : for command input</td>
</tr>
<tr>
<td>re</td>
<td>Simulate smart terminal on dumb terminal</td>
</tr>
<tr>
<td>remap</td>
<td>Accept macros within macros</td>
</tr>
<tr>
<td>report</td>
<td>Show the largest size of changes on status line</td>
</tr>
<tr>
<td>ro</td>
<td>Make file readonly</td>
</tr>
<tr>
<td>scroll=12</td>
<td>Set screen size as 12 lines</td>
</tr>
<tr>
<td>shell=/bin/bash</td>
<td>Set shell escape to /bin/bash</td>
</tr>
<tr>
<td>showmode</td>
<td>Show current mode on status line</td>
</tr>
<tr>
<td>slow</td>
<td>Postpone display updates during inserts</td>
</tr>
<tr>
<td>sm</td>
<td>Show matching parentheses when typing</td>
</tr>
<tr>
<td>sw=8</td>
<td>Set shift width to 8 characters</td>
</tr>
<tr>
<td>tags=/usr/lib/tags</td>
<td>Set path for files checked for tags</td>
</tr>
<tr>
<td>term</td>
<td>Print terminal type</td>
</tr>
<tr>
<td>terse</td>
<td>Print terse messages</td>
</tr>
<tr>
<td>timeout</td>
<td>Eliminate 1-second time limit for macros</td>
</tr>
<tr>
<td>tl=3</td>
<td>Set significance of tags beyond 3 characters (0 = all)</td>
</tr>
<tr>
<td>ts=8</td>
<td>Set tab stops to 8 for text input</td>
</tr>
<tr>
<td>wa</td>
<td>Inhibit normal checks before write commands</td>
</tr>
<tr>
<td>warn</td>
<td>Display the warning message &quot;No write since last change&quot;</td>
</tr>
<tr>
<td>window=24</td>
<td>Set text window as 24 lines</td>
</tr>
<tr>
<td>wm=0</td>
<td>Set automatic wraparound 0 spaces from right margin</td>
</tr>
</tbody>
</table>

Options can also be permanently set by including them in ~/.exrc (Vi) or ~/.vimrc (Vim)
SHOW DATABASES;
USE CompanyDatabase;
SELECT DATABASE();
DROP DATABASE CompanyDatabase;

SHOW TABLES;
CREATE TABLE customers (  
cusid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
firstname VARCHAR(32), lastname VARCHAR(32), dob DATE,  
city VARCHAR(24), zipcode VARCHAR(5));
CREATE TABLE payments (  
payid INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
date DATE, fee INT, bill VARCHAR(128), cusid INT,  
CONSTRAINT FK1 FOREIGN KEY (cusid) REFERENCES customers(cusid));

INSERT INTO customers (firstname,lastname,dob)  
VALUES ('Arthur','Dent',1959-08-01), ('Trillian',';',1971-03-19);
DELETE FROM customers WHERE firstname LIKE 'Zaphod';
UPDATE customers SET city = 'London' WHERE zipcode = 'L1 42HG';
CREATE INDEX lastname_index ON customers(lastname);
ALTER TABLE customers ADD INDEX lastname_index (lastname);

DESCRIBE customers;
SHOW CREATE TABLE customers;
SHOW INDEXES FROM customers;
DROP TABLE customers;

ALTER TABLE customers MODIFY city VARCHAR(32);

CREATE VIEW cust_view AS  
SELECT * FROM customers WHERE city != 'London';

COMMIT;
ROLLBACK;
START TRANSACTION;
BEGIN;

If no database has been selected for use, tables must be referenced by `databasename.tablename`. 
SELECT * FROM customers;

SELECT firstname, lastname FROM customers LIMIT 5;

SELECT firstname, lastname FROM customers LIMIT 1000,5;
SELECT firstname, lastname FROM customers OFFSET 1000 LIMIT 5;

SELECT firstname, lastname FROM customers WHERE zipcode = 'L1 42HG';
SELECT firstname, lastname FROM customers WHERE zipcode IS NOT NULL;

SELECT * FROM customers ORDER BY lastname, firstname;
SELECT * FROM customers ORDER by zipcode DESC;

SELECT firstname, lastname, TIMESTAMPDIFF(YEAR,dob,CURRENT_DATE) AS age FROM customers;
SELECT city, COUNT(*) FROM customers GROUP BY city;

SELECT cusid, SUM(fee) FROM payments GROUP BY cusid;
SELECT cusid, AVG(fee) FROM payments GROUP BY cusid HAVING AVG(fee)<50;

SELECT MAX(fee) FROM payments;
SELECT COUNT(*) FROM customers;
SELECT cusid FROM payments t1 WHERE fee = (SELECT MAX(t2.fee) FROM payments t2 WHERE t1.cusid=t2.cusid);
SELECT @maxfee:=MAX(fee) FROM payments;
SELECT cusid FROM payments t1 WHERE fee = @maxfee;

SELECT * FROM customers WHERE lastname IN (SELECT lastname FROM customers GROUP BY lastname HAVING COUNT(lastname) > 1);

SELECT * FROM customers WHERE firstname LIKE 'Trill%';
SELECT * FROM customers WHERE firstname REGEXP '^Art.*r$';

SELECT firstname, lastname FROM customers WHERE zipcode IS 'L1 42HG'
UNION
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
SELECT firstname, lastname FROM customers WHERE zipcode IS 'L1 42HG'
INTERSECT
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;
SELECT firstname, lastname FROM customers WHERE zipcode IS 'L1 42HG'
EXCEPT
SELECT firstname, lastname FROM customers WHERE cusid > 4242001;

Select all columns from the customers table
Select first and last name of customers, showing 5 records only
Select first and last name of customers, skipping the first 1000 records and showing 5 records only
Select first and last name of customers whose zip code is "L1 42HG"
Select first and last name of customers with an existing zip code
Select customers in alphabetical order by last name, then first name
Select customers, sorting them by zip code in reverse order
Select first name, last name, and calculated age of customers
Show all cities, retrieving each unique output record only once
Show all cities and the number of customers in each city. NULL values are not counted
Show all fee payments grouped by customer ID, summed up
Show the average of fee payments grouped by customer ID, where this average is less than 50
Show the highest fee in the table
Show how many rows are in the table
Show the customer ID that pays the highest fee (via a subquery)
Show the customer ID that pays the highest fee (via a user set variable)
Show the customers which have same last name as other customers
Show the customer IDs that pay fees higher than the highest fee paid by customer ID 4242001

Select customers whose first name matches the expression:
% any number of chars, even zero
_ a single char
Select customers whose first name matches the regex

Select customers that satisfy any of the two requirements
Select customers that satisfy both of the two requirements
Select customers that satisfy the first requirement but not the second
<table>
<thead>
<tr>
<th>SQL</th>
<th>MySQL</th>
<th>Operation</th>
</tr>
</thead>
</table>
| `SELECT customers.name, payments.bill`  
  `FROM customers, payments`  
  `WHERE customers.cusid = payments.cusid;`  
  `SELECT customers.name, payments.bill`  
  `FROM customers, payments`  
  `[ JOIN | INNER JOIN | CROSS JOIN ]`  
  `ON customers.cusid = payments.cusid;`  
  `SELECT customers.name, payments.bill`  
  `FROM customers, payments`  
  `[ JOIN | INNER JOIN | CROSS JOIN ]`  
  `USING (cusid);`  
  `SELECT customers.name, payments.bill`  
  `FROM customers JOIN payments`  
  `[ JOIN | INNER JOIN | CROSS JOIN ]`  
  `USING (cusid);`  
  `SELECT customers.name, payments.bill`  
  `FROM customers CROSS JOIN payments;`  
  `SELECT customers.name, payments.bill`  
  `FROM customers JOIN payments;`  
  `SELECT customers.name, payments.bill`  
  `FROM customers LEFT JOIN payments`  
  `ON customers.cusid = payments.cusid;`  
  `SELECT customers.name, payments.bill`  
  `FROM customers JOIN payments;`  
  `SELECT customers.name, payments.bill`  
  `FROM customers RIGHT JOIN payments`  
  `ON customers.cusid = payments.cusid;` | | Perform a **join** (aka inner join) of two tables to select data that are in a relationship  
  | | Perform a **cross join** (aka Cartesian product) of two tables  
  | | Perform a **left join** (aka left outer join) of two tables, returning records matching the join condition and also records in the left table with unmatched values in the right table  
  | | Perform a **right join** (aka right outer join) of two tables, returning records matching the join condition and also records in the right table with unmatched values in the left table |
MySQL is the most used open source RDBMS (Relational Database Management System). It runs on TCP port 3306. On RHEL 7 and later it is replaced by its fork MariaDB, but the names of the client and of most tools remain unchanged.

mysqld_safe

Start the MySQL server (mysqld) with safety features such as restarting the server if errors occur and logging runtime information to the error logfile. This is the recommended command

mysql_install_db (deprecated)
mysqld --initialize

Initialize the MySQL data directory, create system tables, and set up an administrative account. To be run just after installing the MySQL server

mysql_secure_installation

Set password for root, remove anonymous users, disable remote root login, and remove test database. To be run just after installing the MySQL server

mysql -u root -p

Login to MySQL as root and prompt for the password

mysql -u root -p -password

Login to MySQL as root with the specified password

mysql -u root -p -h host -P port

Login to the specified remote MySQL host and port

mysql -u root -p -eNB'SHOW DATABASES'

Run an SQL command via MySQL. Flags are:
e Run in batch mode
N Do not print table header
B Do not print table decoration characters +-|

mysqldump -u root -p --all-databases > dump.sql

Backup all databases to a dump file

mysqldump -u root -p db > dump.sql

Backup a database to a dump file

mysqldump -u root -p --databases db1 db2 > dump.sql

Backup multiple databases to a dump file

mysql -u root -p db table1 table2 > dump.sql

Backup some tables of a database to a dump file

mysql -u root -p db < dump.sql

Restore all databases from a dump file (which contains a complete dump of a MySQL server)

mysql -u root -p db < dump.sql

Restore a specific database from a dump file (which contains one database)

mysql_upgrade -u root -p

Check all tables in all databases for incompatibilities with the current version of MySQL

mysqlcheck

Perform table maintenance. Each table is locked while is being processed. Options are:
--check Check table for errors (default)
--analyze Analyze table
--optimize Optimize table
--repair Repair table; can fix almost all problems except unique keys that are not unique

mysqlcheck --check db table

Check the specified table of the specified database

mysqlcheck --check --databases db1 db2

Check the specified databases

mysqlcheck --check --all-databases

Check all databases
MySQL - tools

mysqlslap
Tool for MySQL stress tests

mysqltuner.pl
Review the current MySQL installation configuration for performances and stability

mysqlreport  (obsolete)
Generate a user-friendly report of MySQL status values

mytop
Monitor MySQL processes and queries

innotop
Monitor MySQL InnoDB transactions

dbs="$(mysql -uroot -p$password -Bse'SHOW DATABASES;';")"
for db in $dbs
do
  [operation on $db]
done

Perform an operation on each database name
SELECT Host, User FROM mysql.user;
CREATE USER 'user'@'localhost' IDENTIFIED BY 'p4ssw0rd';
DROP USER 'user'@'localhost';
SET PASSWORD FOR 'user'@'localhost' = PASSWORD('p4ssw0rd');
SET PASSWORD FOR 'user'@'localhost' = '*7E684A3DF6273CD1B6DE53';
SHOW GRANTS FOR 'user'@'localhost';
GRANT ALL PRIVILEGES ON database.* TO 'user'@'localhost';
REVOKE ALL PRIVILEGES ON database.* FROM 'user'@'localhost';
GRANT SELECT ON database.* TO 'john'@'localhost' IDENTIFIED BY 'p4ssw0rd';
GRANT SELECT ON database.* IDENTIFIED BY PASSWORD '*7E684A3DF6273CD1B6DE53';
FLUSH PRIVILEGES;
SELECT * INTO OUTFILE 'file.csv'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '
' FROM database.table;
USE database; SOURCE dump.sql;
USE database; LOAD DATA LOCAL INFILE 'file' INTO TABLE table;
DO SLEEP(n);
SELECT SLEEP(n);
SET PROFILING=1;
SHOW PROFILE;
SELECT /*!99999 comment*/ * FROM database.table;
SELECT /*!v statement*/ * FROM database.table;
\c
\! command
TEE logfile

List all MySQL users
Create a MySQL local user and set their password
Delete a MySQL user
Set a password for a MySQL user.
The password can be specified either in plaintext or by its hash value
Show permissions for a user
Grant permissions to a user
Revoke permissions from a user; must match the already granted permission on the same database or table
Create a MySQL user and set their grants at the same time
Reload and commit the grant tables; must be run after any GRANT command
Export a table to a CSV file
Restore a database from a dump file
Populate a table with data from a file (one record per line, values separated by tabs)
Sleep for n seconds
Enable profiling
Show the profile of the last executed query, with detailed steps and their timing
Send an SQL statement to the server
Display result in vertical format, showing each record in multiple rows
Insert a comment
The commented statement is executed only if MySQL is version v or higher
Cancel current input
Run a shell command
Log all I/O of the current MySQL session to the specified logfile
SHOW VARIABLES;
SHOW SESSION VARIABLES;
SHOW LOCAL VARIABLES;
SHOW GLOBAL VARIABLES;
SHOW VARIABLES LIKE '%query%';
SHOW VARIABLES LIKE 'hostname';
SELECT @@hostname;

SET sort_buffer_size=10000;
SET SESSION sort_buffer_size=10000;
SET LOCAL sort_buffer_size=10000;
SET @@session.sort_buffer_size=10000;
SET @@local.sort_buffer_size=10000;
SET GLOBAL sort_buffer_size=10000;
SET @@global.sort_buffer_size=10000;

SHOW STATUS;
SHOW SESSION STATUS;
SHOW LOCAL STATUS;
SHOW GLOBAL STATUS;
SHOW STATUS LIKE '%wsrep%';

SHOW WARNINGS;
SHOW ERRORS;

SHOW TABLE STATUS;
SHOW ENGINE INNODB STATUS;
SELECT * FROM information_schema.processlist;
SHOW FULL PROCESSLIST;
SELECT * FROM information_schema.processlist WHERE user='you';

SHOW CREATE TABLE table;
SHOW CREATE VIEW view;

SELECT VERSION();

SELECT CURDATE();
SELECT CURRENT_DATE;
SELECT CURTIME();
SELECT CURRENT_TIME;
SELECT NOW();

SELECT USER();

\s
SELECT table_schema AS "Name",
SUM(data_length+index_length)/1024/1024 AS "Size in Mb"
FROM information_schema.tables GROUP BY table_schema;

Display the sizes of all databases in the system (counting data + indexes)

SELECT table_schema AS "Name",
SUM(data_length+index_length)/1024/1024 AS "Size in Mb"
FROM information_schema.tables WHERE table_schema='database';

Display the size of database

SELECT table_name AS "Name",
ROUND(((data_length)/1024/1024),2) AS "Data size in Mb",
ROUND(((index_length)/1024/1024),2) AS "Index size in Mb"
FROM information_schema.tables WHERE table_schema='database'
ORDER BY table_name;

Display data and index size of all tables of database

SELECT table_name, table_rows
FROM information_schema.tables WHERE table_schema='database';

Print an estimate of the number of rows of each table of database

SELECT SUM(data_length+index_length)/1024/1024 AS "InnoDB Mb"
FROM information_schema.tables WHERE engine='InnoDB';

Display the amount of InnoDB data in all databases

SELECT table_name, engine
FROM information_schema.tables WHERE table_schema = 'database';

Print name and engine of all tables in database

SELECT CONCAT('KILL ',id,';')
FROM information_schema.processlist WHERE user='user'
INTO OUTFILE '/tmp/killuser'; SOURCE /tmp/killuser;

Kill all connections belonging to user

SELECT COUNT(1) SlaveThreadCount
FROM information_schema.processlist WHERE user='system user';

Distinguish between master and slave server; returns 0 on a master, >0 on a slave

SELECT ROUND(SUM(CHAR_LENGTH(field)<40)*100/COUNT(*) ,2)
FROM table;

Display the percentage of rows on which the string field is shorter than 40 chars

SELECT CHAR_LENGTH(field) AS Length, COUNT(*) AS Occurrences
FROM table GROUP BY CHAR_LENGTH(field);

Display all different lengths of string field and the number of times they occur

SELECT MAX(CHAR_LENGTH(field)) FROM table;

Display the longest string stored in field

SHOW FULL TABLES IN database WHERE table_type LIKE 'VIEW';

Display the list of views in database

SELECT "Table 1" AS `set`, t1.* FROM table1 t1 WHERE
ROW(t1.col1, t1.col2, t1.col3) NOT IN (SELECT * FROM table2)
UNION ALL
SELECT "Table 2" AS `set`, t2.* FROM table2 t2 WHERE
ROW(t2.col1, t2.col2, t2.col3) NOT IN (SELECT * FROM table1);

Display the differences between the contents of two tables table1 and table2 (assuming the tables are composed of 3 columns each)
How to resync a master-slave replication

1. On the master, on terminal 1:
   ```
   mysql -uroot -p
   RESET MASTER;
   FLUSH TABLES WITH READ LOCK;
   SHOW MASTER STATUS;
   Note the values of MASTER_LOG_FILE and MASTER_LOG_POS; these values will need to be copied on the slave
   ```

2. On the master, on terminal 2:
   ```
   mysqldump -uroot -p --all-databases > /path/to/dump.sql
   It is not necessary to wait until the dump completes
   ```

3. On the master, on terminal 1:
   ```
   UNLOCK TABLES;
   ```

4. Transfer the dump file from the master to the slave

5. On the slave:
   ```
   mysql -uroot -p
   STOP SLAVE;
   SOURCE /path/to/dump.sql;
   RESET SLAVE;
   CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.nnnnnn', MASTER_LOG_POS=mm;
   START SLAVE;
   SHOW SLAVE STATUS;
   ```

How to recover the MySQL root password

1. Stop the MySQL server

2. Restart the MySQL server skipping the grant tables
   ```
   mysqld_safe --skip-grant-tables --skip-networking &
   ```

3. Connect to the MySQL server passwordlessly
   ```
   mysql -uroot
   ```

4. Reload the grant tables
   ```
   FLUSH PRIVILEGES;
   ```

5. Change the root password
   ```
   SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpassword');
   ```

6. Stop the MySQL server and restart it normally
PostgreSQL (aka Postgres) is an open source object-relational database. By default it listens for connections on TCP port 5432.

```
\list
\l
\list+
\l+
\connect database
\c database
\q
```

List all databases
List all databases, displaying database size and description
Connect to `database`
Quit

**How to set up PostgreSQL with a database owned by a user**

1. Set up PostgreSQL
   
   `postgresql-setup initdb`

2. Change the password of the postgres shell user
   
   `passwd postgres`

3. Create the user shell user
   
   `useradd user`

4. Switch to the postgres shell user and connect to PostgreSQL
   
   `su - postgres`
   `psql -U postgres`

5. In PostgreSQL, create the user
   
   `CREATE ROLE user WITH LOGIN;`
   `\password user`
   `\q`

6. Create a database owned by user
   
   `createdb -E utf8 -l C -T template0 database -O user`

7. Switch to the postgres shell user and connect to PostgreSQL
   
   `su - postgres`
   `psql -U postgres`

8. In PostgreSQL, grant the necessary privileges on database to user
   
   `GRANT ALL PRIVILEGES ON DATABASE database TO user;`
   `\q`

9. Verify that user can login to PostgreSQL
   
   `su - user`
   `psql -U user -W`
The **X Window System** (aka **X11** or **X**) is a windowing system for Linux and UNIX-like OSes, providing a basic framework for GUI applications via a client-server model. A display manager provides a login screen to enter an X session and introduces the user to the desktop environment (e.g. GNOME, KDE, CDE, Enlightenment).

<table>
<thead>
<tr>
<th>Display Manager</th>
<th>Configuration files</th>
<th>Display Manager greeting screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>xdm X Display Manager</td>
<td><code>/etc/x11/xdm/Xaccess</code></td>
<td>Control inbound requests from remote hosts</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xresources</code></td>
<td>Configuration settings for X applications and the login screen</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xservers</code></td>
<td>Association of X displays with local X server software, or with X terminals via XDMCP</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xsession</code></td>
<td>Script launched by xdm after login</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/Xsetup_0</code></td>
<td>Script launched before the graphical login screen</td>
</tr>
<tr>
<td></td>
<td><code>/etc/x11/xdm/xdm-config</code></td>
<td>Association of all xdm configuration files</td>
</tr>
<tr>
<td>gdm GNOME Display Manager</td>
<td><code>/etc/gdm/gdm.conf</code> or <code>/etc/gdm/custom.conf</code></td>
<td>Defined in <code>/etc/x11/xdm/Xresources</code> by the line: xlogin*greeting: \ Debian GNU/Linux (CLIENTHOST)</td>
</tr>
<tr>
<td>kdm KDE Display Manager</td>
<td><code>/etc/kde/kdm/kdmrc</code></td>
<td>Configured via kdm_config</td>
</tr>
</tbody>
</table>

```
/etc/init.d/xdm start
/etc/init.d/gdm start
/etc/init.d/kdm start
```
Start the appropriate Display Manager

```
xorgconfig  # (Debian)
Xorg -configure  # (Red Hat)
xorgcfg  # (Debian)
system-config-display  # (Red Hat)
X -version
xdpyinfo
xwininfo
xhost + 10.3.3.3
xhost - 10.3.3.3
switchdesk gde
gnome-shell --version
```
Configure X (text mode)

```
xorgconf   # (Debian)
xorgcfg    # (Red Hat)
```
Configure X (graphical mode)

Show which version of X is running

```
Display information about the X server
Display information about windows
Add or remove 10.3.3.3 to the list of hosts allowed making X connections to the local machine
Switch to the GDE Display Manager at runtime
Show which version of GNOME is running
```

```
/etc/X11/xorg.conf
~/.Xresources
$DISPLAY
```
Configuration file for X

```
Configuration settings for X applications, in the form
program*resource: value
```

```
Environment variable defining the display name of the X server, in the form
hostname:displaynumber.screennumber
```

The following line in `/etc/inittab` instructs init to launch XDM at runlevel 5:

```
x:5:respawn:/usr/X11R6/bin/xdm -nodaemon
```

The following lines in `/etc/sysconfig/desktop` define GNOME as the default Display Environment and Display Manager:

```
desktop="gde"
displaymanager="gdm"
```
xdotool
xdotool getwindowfocus
xdotool selectwindow
xdotool key --window 12345678 Return

xdotool
X automation tool
Get the ID of the currently focused window (if run in command line, it is the terminal where this command is typed)
Pop up an X cursor and get the ID of the window selected by it
Simulate a \(\text{RETURN}\) keystroke inside window ID 12345678

xprop
xprop | grep WM_CLASS

xprop
X property displayer. Pops up a cursor to select a window
Get process name and GUI application name of the selected window

xrandr
xrandr -q
xrandr --output eDP1 --right-of VGA1

xrandr
Configuration utility for the RandR (Resize and Rotate) X extension
Show screen(s) size and resolution
Extend the screen on an additional VGA physical monitor situated to the left

xsel
xsel -b < file
xsel -b -a < file
xsel -b -o

xsel
Manipulate the X selection (primary, secondary, and clipboard)
Copy the contents of a file to the X clipboard
Append the contents of a file to the X clipboard
Output onscreen the contents of the X clipboard

cat file | xclip -i

cat file | xclip -i
Copy the contents of a file to the X clipboard

xosview

xosview
Monitor able to display several system parameters (CPU usage, memory usage, load average, page swapping, interrupts, battery level, etc.)

mkfontdir
mkfontdir
Catalog the newly installed fonts in the new directory

xset fp+ /usr/local/fonts
xset fp+ /usr/local/fonts
Dynamically add new installed fonts in /usr/local/fonts to the X server
Start the X font server

fc-cache
fc-cache
Install fonts and build font information cache

fc-scan fontfile.ttf
fc-scan fontfile.ttf
Get information about a font
### X Window - keysim codes

<table>
<thead>
<tr>
<th>Main</th>
<th>Latin 1</th>
<th>Latin 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackSpace</td>
<td>ff08</td>
<td>space</td>
</tr>
<tr>
<td>Tab</td>
<td>ff09</td>
<td>exclam</td>
</tr>
<tr>
<td>Linefeed</td>
<td>ff0a</td>
<td>quotedbl</td>
</tr>
<tr>
<td>Clear</td>
<td>ff0b</td>
<td>numbersign</td>
</tr>
<tr>
<td>Return</td>
<td>ff0d</td>
<td>percent</td>
</tr>
<tr>
<td>Pause</td>
<td>ff13</td>
<td>amperandslash</td>
</tr>
<tr>
<td>Scroll_Lock</td>
<td>ff14</td>
<td>apostrophe</td>
</tr>
<tr>
<td>Sys_Req</td>
<td>ff15</td>
<td>quotationmark</td>
</tr>
<tr>
<td>Escape</td>
<td>ff1b</td>
<td>parenleft</td>
</tr>
<tr>
<td>Delete</td>
<td>fffe</td>
<td>parenright</td>
</tr>
</tbody>
</table>

### Cursor control

| Home | ff50 | plus | 002b | Edieresis | 00cb | ogonek | 01b2 |
| Left | ff51 | comma | 002c | Igrave | 00cc | Istroke | 01b3 |
| Up | ff52 | minus | 002d | Iacute | 00cd | Icaron | 01b5 |
| Right | ff53 | period | 002e | Icirumflex | 00ce | Iacute | 01b6 |
| Down | ff54 | slash | 002f | Idiaeresis | 00cf | caron | 01b7 |
| Prior | ff55 | 0 - 9 | 0030 - 0039 | ETH | 00d0 | scaron | 01b9 |
| Page_Up | ff55 | colon | 003a | Eth | 00d0 | Ascender | 01ba |
| Next | ff56 | semicolon | 003b | Ntilde | 00d1 | Tcaron | 01bb |
| Page_Down | ff56 | less | 003c | Ograve | 00d2 | Zacute | 01bc |
| End | ff57 | equal | 003d | Oacute | 00d3 | Doubleacute | 01bd |
| Begin | ff58 | greater | 003e | Ocirumflex | 00d4 | Zcaron | 01be |

### Misc functions

| Select | ff60 | question | 003f | Otilde | 00d5 | Zabovedot | 01bf |
| Print | ff61 | A - 2 | 0041 | multiply | 0046 | Aacute | 01c0 |
| Execute | ff62 | Aabreve | 0047 | Abar | 0048 | Agrave | 01c3 |
| Insert | ff63 | bracketleft | 005b | Oslash | 0048 | Lacron | 01c5 |
| Undo | ff65 | bracketslash | 005c | Ooblique | 0048 | Cacute | 01c6 |
| Redo | ff66 | bracketright | 005d | Ugrave | 0049 | Ccaron | 01c8 |
| Menu | ff67 | asciicircumflex | 005e | Uacute | 00da | Eogonek | 01ca |
| Find | ff68 | underscorescore | 005f | Ucircumflex | 00db | Escaron | 01cc |
| Cancel | ff69 | grave | 0060 | Udiaeresis | 00dc | Dcaron | 01cf |
| Help | ff6a | a - z | 0061 - 007a | THORN | 00de | Nacute | 01d1 |
| Break | ff6b | braceleft | 007b | Thor | 00de | Ncaron | 01d2 |
| Mode_switch | ff7e | braceleft | 007b | braceright | 007d | agrave | 00e0 | Rcaron | 01d4 |
| script_switch | ff7e | braceslash | 007c | acute | 00e1 | Uring | 01d9 |
| Num_Lock | ff7f | acent | 00e2 | adieresis | 00e3 | Racute | 01e0 |

### Modifiers

| Shift_L | ff1 | sterling | 00a3 | aring | 00e5 | Agrave | 01e3 |
| Shift_R | ff2 | currency | 00a4 | ae | 00e6 | lacute | 01e5 |
| Control_L | ff3 | yen | 00a5 | ccedilla | 00e7 | caacute | 01e6 |
| Control_R | ff4 | brokenbar | 00a6 | egrave | 00e8 | Scarone | 01e8 |
| Caps_Lock | ff5 | section | 00a7 | eacute | 00e9 | Eogonek | 01e9 |
| Shift_Lock | ff6 | copyright | 00a9 | Edieresis | 00ea | Ecaron | 01ec |
| Meta_L | ff7 | ordfeminine | 00aa | edieresis | 00eb | Dcaron | 01ef |
| Meta_R | ff8 | guillemotleft | 00ab | igraeve | 00ec | Dstroke | 01f0 |
| Alt_L | ffe9 | guillemotright | 00ac | iacute | 00ed | Dacute | 01f1 |
| Alt_R | ffea | nosign | 00ad | idiaeresis | 00ef | Odoublacute | 01f2 |
| Super_L | ffeb | registered | 00ae | eth | 00f0 | Rcaron | 01f8 |
| Super_R | ffec | registered | 00ae | eth | 00f0 | Rcaron | 01f8 |
| Hyper_L | ffe3 | macron | 00af | ntilde | 00f1 | uring | 01f9 |
| Hyper_R | ffe4 | degree | 00b0 | ogave | 00f2 | Udoublacute | 01f2f |
| Hyper_L | ffe6 | plusminus | 00b1 | oacute | 00f3 | Tcedilla | 01fe |
| Hyper_R | ffe8 | twosuperior | 00b2 | ocirumflex | 00f4 | abovebot | 01ff |
| threegrave | 00b3 | threesuperior | 00b3 | otilde | 00f5 | Oacute | 01f6 |
| acute | 00b4 | acute | 00b4 | odiareresis | 00f6 | Scarone | 01f8 |
| mus | 00b5 | division | 00b7 | oslash | 00f8 | Scarone | 01f8 |
| paragraph | 00b6 | periodcentered | 00b7 | oblique | 00f8 | Scarone | 01f8 |
| cedilla | 00b8 | cedilla | 00b8 | ugrave | 00f9 | Scarone | 01f8 |
| onesuperior | 00b9 | onesuperior | 00b9 | uacute | 00fa | Scarone | 01f8 |
| macrue | 00ba | macrue | 00ba | ucircumflex | 00fb | Scarone | 01f8 |
| guillemotright | 00bb | guillemotright | 00bb | udiaeresis | 00fc | Scarone | 01f8 |
| onequarter | 00bc | onequarter | 00bc | ycute | 00fd | Scarone | 01f8 |
| onehalf | 00bd | onehalf | 00bd | thorn | 00fe | Scarone | 01f8 |
| threequarters | 00be | threequarters | 00be | ydiaeresis | 00ff | Scarone | 01f8 |

This table is derived from keysymdef.h which defines keysym codes (i.e. characters or functions associated with each key in the X Window System) as XK_key and its hex value. The key can be passed as argument to the xdotoool key command.

© Daniele Raffo  
dr0.ch
## /etc/passwd

<table>
<thead>
<tr>
<th>Login name</th>
<th>Hashed password</th>
<th>UID</th>
<th>GID</th>
<th>GECOS field</th>
<th>Home directory</th>
<th>Login shell</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>x:0:0:/root:/bin/bash</td>
<td>0</td>
<td>0</td>
<td></td>
<td>/root</td>
<td>/bin/bash</td>
</tr>
<tr>
<td>bin</td>
<td>x:1:1:/bin:/bin/bash</td>
<td>1</td>
<td>1</td>
<td></td>
<td>/bin</td>
<td>/bin/bash</td>
</tr>
<tr>
<td>jdoe</td>
<td>x:500:100:John Doe,,555-1234,,:/home/jdoe:/bin/bash</td>
<td>500</td>
<td>100</td>
<td></td>
<td>/home/jdoe</td>
<td>/bin/bash</td>
</tr>
</tbody>
</table>

1. Login name
2. Hashed password (obsolete), or x if password is in /etc/shadow
3. UID – User ID
4. GID – Default Group ID
5. GECOS field – Information about the user: Full name, Room number, Work phone, Home phone, Other
6. Home directory of the user
7. Login shell (if set to /sbin/nologin or /bin/false, user will be unable to log in)

## /etc/shadow

<table>
<thead>
<tr>
<th>Login name</th>
<th>Hashed password</th>
<th>Date of last password change</th>
<th>Days before password may be changed</th>
<th>Days after which password must be changed</th>
<th>Days before password expiration that user is warned</th>
<th>Days after password expiration that account is disabled</th>
<th>Date of account disabling</th>
<th>Reserved field</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>*:15637:0:99999:7:::</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>jdoe</td>
<td>!$6$YOiH1otQ$KxeeUKHEXK8e3jCUdw9Rxy3Wu53:15580:0:99999:7::15766:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Login name
2. Hashed password (* if account is disabled, ! if no password is set, prefixed by !! if the account is locked).
3. Composed of the following subfields separated by $:
   a. Hashing algorithm: 1 = MD5, 2a = Blowfish, 5 = SHA256, 6 = SHA512 (recommended)
   b. Random salt, up to 16 chars long. This is to thwart password cracking attempts based on rainbow tables
   c. String obtained by hashing the user's plaintext password concatenated to the stored salt
4. Date of last password change (in number of days since 1 January 1970)
5. Days before password may be changed; if 0, user can change the password at any time
6. Days after which password must be changed
7. Days before password expiration that user is warned
8. Days after password expiration that account is disabled
9. Date of account disabling (in number of days since 1 January 1970)

## /etc/group

<table>
<thead>
<tr>
<th>Group name</th>
<th>Encrypted password</th>
<th>GID</th>
<th>Group members</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>x:root</td>
<td>0</td>
<td>root</td>
</tr>
<tr>
<td>jdoe</td>
<td>x:501</td>
<td>100</td>
<td>jdoe, asmith</td>
</tr>
</tbody>
</table>

1. Group name
2. Encrypted password, or x if password is in /etc/gshadow
3. GID – Group ID
4. Group members (if this is not their Default Group)

## /etc/gshadow

<table>
<thead>
<tr>
<th>Group name</th>
<th>Encrypted password</th>
<th>GID</th>
<th>Group administrators</th>
<th>Group members</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>::root</td>
<td>0</td>
<td></td>
<td>root, jdoe</td>
</tr>
<tr>
<td>jdoe</td>
<td>!:</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>staff</td>
<td>0cfz7IpLhW19i::root, jdoe</td>
<td>530</td>
<td>jdoe, asmith</td>
<td></td>
</tr>
</tbody>
</table>

1. Group name
2. Encrypted password, or ! if no password is set (default)
3. Group administrators
4. Group members

/etc/shadow and /etc/gshadow are mode 000 and therefore readable only by the root user.
useradd -m user
Create a user account, creating and populating their homedir from /etc/skel

useradd -mc "Name Surname" user
Create a user account, specifying their full name

useradd -ms /bin/ksh user
Create a user account, specifying their login shell

useradd -D
Show default values for user account creation, as specified in /etc/login.defs and /etc/default/useradd

usermod -c "Name Surname" user
Modify the GECOS field of a user account

usermod -L user
Lock a user account

usermod -U user
Unlock a user account

Most options for usermod and useradd are the same.

userdel -r user
Delete a user and their homedir

chfn user
Change the GECOS field of a user

chsh user
Change the login shell of a user

passwd user
Change the password of a user

passwd -l user
Lock a user account

passwd -S user
Show information about a user account: username, account status (L=locked, P=password, NP=no password), date of last password change, min age, max age, warning period, inactivity period in days

chage -E 2022-02-14 user
Change the password expiration date; account will be locked at that date

chage -d 13111 user
Change the date (in number of days since 1 January 1970) of last password change

chage -d 0 user
Force the user to change password at their next login

chage -M 30 user
Change the max number of days during which a password is valid

chage -m 7 user
Change the min number of days between password changes

chage -W 15 user
Change the number of days before password expiration that the user will be warned

chage -I 3 user
Change the number of days after password expiration before the account is locked

chage -l user
List password aging information for a user

chpasswd
Tool for batch update of passwords. Reads from stdin a list of username:password

vipw
Edit manually /etc/passwd, /etc/shadow, /etc/group, or /etc/gshadow

vigr

adduser
deluser
User-friendly front-end commands for user management

system-config-users (Red Hat)
GUI for user and group management
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>groupadd group</code></td>
<td>Create a group</td>
</tr>
<tr>
<td><code>groupmod -n newname oldname</code></td>
<td>Change a group name</td>
</tr>
<tr>
<td><code>groupdel group</code></td>
<td>Delete a group</td>
</tr>
<tr>
<td><code>gpasswd group</code></td>
<td>Set or change the password of a group</td>
</tr>
<tr>
<td><code>gpasswd -a user group</code></td>
<td>Add a user to a group</td>
</tr>
<tr>
<td><code>gpasswd -d user group</code></td>
<td>Delete a user from a group</td>
</tr>
<tr>
<td><code>gpasswd -A user group</code></td>
<td>Add a user to the list of administrators of the group</td>
</tr>
<tr>
<td><code>addgroup</code></td>
<td>User-friendly front-end commands for group management</td>
</tr>
<tr>
<td><code>delgroup</code></td>
<td></td>
</tr>
</tbody>
</table>
On a system, every user is identified by a numeric **UID (User ID)**, and every group by a numeric **GID (Group ID)**. UID 0 is assigned to the superuser.

UIDs from 0 to 99 should’ be reserved for static allocation by the system and not be created by applications. UIDs from 100 to 499 should’ be reserved for dynamic allocation by the superuser and post-install scripts. UIDs for user accounts start from 500 (Red Hat) or 1000 (SUSE, Debian).

’ as recommended by the Linux Standard Base core specifications

A process has an effective, saved, and real UID and GID.

<table>
<thead>
<tr>
<th><strong>Effective UID</strong></th>
<th>Used for most access checks, and as the owner for files created by the process. An unprivileged process can change its effective UID only to either its saved UID or its real UID.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Saved UID</strong></td>
<td>Used when a process running with elevated privileges needs to temporarily lower its privileges. The process changes its effective UID (usually root) to an unprivileged one, and its privileged effective UID is copied to the saved UID. Later, the process can resume its elevated privileges by resetting its effective UID back to the saved UID.</td>
</tr>
<tr>
<td><strong>Real UID</strong></td>
<td>Used to identify the real owner of the process and affect the permissions for sending signals. An unprivileged process can signal another process only if the sender’s real or effective UID matches the receiver’s real or saved UID. Child processes inherit the credentials from the parent, so they can signal each other.</td>
</tr>
</tbody>
</table>

`whoami`  
Print your username (as effective UID)

`id`  
Print your real and effective UID and GID, and the groups of which you are a member

`id user`  
Print real and effective UID and GID, and group membership information, about *user*

`id -u`  
Print your effective UID

`who`  
Print the list of users logged into the system

`w`  
Print the list of users logged into the system, and what they are doing

`users`  
Print names of currently logged in users

`finger user@host`  
Print information about *user* on *host*

`rwho`  
Print information about currently logged in users for all hosts on the local network

`rusers`  
Print names of currently logged in users for all hosts on the local network
Sudo is a mechanism that allows running a command as another user. Sudo access rights are defined in the sudoers files /etc/sudoers and /etc/sudoers.d/*; these files must be edited only via visudo. Commands run by sudo users are logged via syslog on /var/log/auth.log (Debian) or /var/log/secure (Red Hat).

**Sudo**

- `sudo -u user command`  
  Run command as user

- `sudo command`  
  Run command as root

- `sudo -u root command`  
  Run command as root

- `sudo su -`  
  Login on an interactive shell as root

- `sudo -i`  
  Login as root with a shell, even if the user's shell is /sbin/nologin or similar

- `sudo -u user -s`  
  List the allowed commands for the current user

- `sudo -l`  
  Run again the last command, but this time as root

**Sudoedit**

- `sudoedit /etc/passwd`  
  Edit safely a file (in this case, /etc/passwd) according to security policies. It is recommended to have nonprivileged users run this command instead of sudoing text editors as root on protected files, because the text editor might spawn a shell, causing security issues

- `sudo -e /etc/passwd`  
  Edit safely the sudoers file

**Visudo**

- `visudo`  
  Check the sudoers file for syntax errors, unused aliases, etc.

- `visudo -c`  
  Check the sudoers file for syntax errors, unused aliases, etc.

**Su**

- `su user`  
  Run a shell as user

- `su`  
  Run a shell as root

- `su root`  
  Run a shell as root

- `su -`  
  Ensure that the spawned shell is a login shell, hence running login scripts and setting the correct environment variables. Recommended option

- `su -c "fdisk -l"`  
  Ensure that the spawned shell is a login shell, hence running login scripts and setting the correct environment variables. Recommended option

**Gksudo**

- `gksudo -u root command`  
  GUI front-ends to su and sudo used to run an X Window command or application as root. Pops up a requester prompting the user for root's password

- `gksu -u root -l`  
  GUI front-ends to su and sudo used to run an X Window command or application as root. Pops up a requester prompting the user for root's password

**Runuser**

- `runuser -u user command`  
  Run command as user. Can be launched only by root
Terminals

chvt n

Make /dev/tty n the foreground terminal

vlock

Lock the virtual console (terminal)

tty

Print your terminal device (e.g. /dev/tty1, /dev/pts/1)

stty

Change or display terminal line settings

stty -ixon

Disable XON/XOFF flow control

clear

Clear the terminal screen

tmux

Terminal multiplexer

reptyr

Attach an existing running program to a new terminal

screen

Screen manager that multiplexes a single virtual VT100/ANSI terminal between multiple processes or shells. When the connection to a terminal is lost (e.g. because the terminal is closed manually, the user logs out, or the remote SSH session goes into timeout), a SIGHUP is sent to the shell and from there to all running child processes which are therefore terminated. The `screen` command starts an interactive shell screen session, to which the user will be able to reattach later.

screen -S sessionname

Start a screen session with the specified session name

screen command

Start the specified command in a screen session; session will end when the command exits

screen -list

Show the list of detached screen sessions

screen -r pid.tty.host

Resume a detached screen session

screen --r owner/pid.tty.host

Resume the last detached screen session

screen -d -R sessionname

Detach a remote screen session and reattach your current terminal to it

How to detach an already running job that was not started in a screen session

(this procedure detaches the job from its parent shell, so that the job will not be killed when the terminal is closed)

1. **CTRL** + **Z**

   Suspend the job

2. bg

   Send the job to background

3. jobs

   Show the number (let us assume is n) of the backgrounded job

4. disown -h %n

   Mark job n so it will not receive a SIGHUP from its parent shell

or

1. screen

   Start a new screen session

2. reptyr pid

   Attach the job with process ID pid to the new terminal (screen session)
**Messaging**

write user

echo "Message" | write user

wall

echo "Message" | wall

talk user

mesg

mesg y
chmod g+w $(tty)

mesg n
chmod g-w $(tty)

Write interactively a message to the terminal of *user* (which must be logged in)

Write a message to the terminal of *user* (which must be logged in)

Write interactively a message to the terminal of all logged in users

Write a message to the terminal of all logged in users

Open an interactive chat session with *user* (which must be logged in)

Display your current message permission status.
This command works by enabling/disabling the group write permission of your
terminal device, which is owned by system group *tty*.
The root user is always able to message users, regardless of their permission status

Allow the other users to message you via *write*, *wall*, and *talk*

Disallow the other users to message you via *write*, *wall*, and *talk*
cron is a job scheduler, allowing repeated execution of commands specified in crontab files.
The crond daemon checks the crontab files every minute and runs the command as the specified user at the specified times. It is not necessary to restart crond after the modification of a crontab file, as the changes will be reloaded automatically.
On Systemd-based distros, Systemd timers are an alternative to cron jobs.

If /etc/cron.allow exists, only users listed therein can access the service.
If /etc/cron.deny exists, all users except those listed therein can access the service.
If none of these files exist, all users can access the service.

/etc/crontab
/etc/cron.d/*
/etc/cron.hourly/
/etc/cron.daily/
/etc/cron.weekly/
/etc/cron.monthly/
/var/spool/cron/user

System-wide crontab files
Scripts placed in these directories will be automatically executed on the specified periods
Crontab of user. This file has the same format as the system-wide crontab files, except that the "user" field is not present

`crontab -e`  Edit your user crontab file
`crontab -l`  List the contents of your crontab file
`crontab -e -u user`  Edit the crontab file of another user (command available only to the superuser)

```
# m  h  dom  mon  dow  user  command
25  6   *   *   1   root  /opt/script1.sh
*/5 16   *   *   *   root  /opt/script2.sh
0,30 7   25  12   *   jdoe  /home/jdoe/foo.sh
3 17   *   1-5  root  /bin/rm /tmp/abc.o
```
every Monday at 6:25 AM
from 4:00 to 4:55 PM every 5 minutes every day
at 7:00 and 7:30 AM on 25th December
at 5:03 PM every day, from Monday to Friday

m  minutes
h  hours
dom  day of month (1-31)
mon  month (1-12 or jan-dec)
dow  day of week (0-7 or sun-sat; 0=7=Sunday)
user  User as whom the command will be executed
command  Command that will be executed at the specified times

The crond daemon also runs anacron jobs, which allow execution of periodic jobs on a machine that is not always powered on, such as a laptop. Only the superuser can schedule anacron jobs, which have a granularity of one day (vs one minute for cron jobs).

/var/spool/anacron/jobid  Date of the last execution of the anacron job identified by jobid

```
# period  delay  job-identifier  command
7       10     cron.weekly     /opt/script3.sh
```
If the job has not been run in the last 7 days, wait 10 minutes and then execute the command

period  period, in days, during which the command was not executed
delay  delay to wait, in minutes, before execution of the command
job-identifier  job identifier in anacron messages; should be unique for each anacron job
command  command that will be executed
at is used for scheduled execution of commands that must run only once. Execution of these commands is the duty of the atd daemon.

If /etc/at.allow exists, only users listed therein can access the service.
If /etc/at.deny exists, all users except those listed therein can access the service.
If none of these files exist, no user except the superuser can access the service.

```
at 5:00pm tomorrow script.sh
at -f listofcommands.txt 5:00pm tomorrow
echo "rm file" | at now+2 minutes
at -l
atq
at -d 3
atrm 3
```

Execute a command once at the specified time (absolute or relative)

List the scheduled jobs

Remove job number 3 from the list

batch

Schedule execution of a command for when the system is not too charged. Reads a command from stdin and runs it when the system's load average falls below 0.8
bc Calculator

dc Calculator featuring unlimited precision arithmetic

factor Find the prime factors of an integer

units Convert quantities between different units

datamash Perform numeric operations, apply statistical functions, or change formatting on tabular data

dv VisiData, an interactive tool to explore and operate on tabular data

jp Draw simple plots from CSV or JSON data
daff Compare tabular data and find the differences

gnuplot Utility to plot 2D and 3D graphs

in2csv Convert various tabular data formats into CSV. Part of the csvkit Python package

sql2csv Execute SQL queries on a database and output the result as CSV

csvclean Correct common syntax error on a CSV file

csvcut Filter data on a CSV file

csvgrep Find data in specific columns of a CSV file

csvjoin Join CSV tables (similarly to SQL JOIN)
csvsort Sort data on a CSV file
csvstack Stack up rows from CSV files
csvformat Convert a CSV file to another format
csvjson Convert a CSV file to JSON format
csvlook Render a CSV file in the terminal as a fixed width table (compatible with Markdown)
csvpy Load a CSV file into a Python CLI shell
csvsql Generate SQL queries from a CSV file and execute them on a database
csvstat Print statistics for all columns of a CSV file

q Execute SQL queries against CSV files
textql Execute SQL queries against CSV files
Utilities - misc

- cal: Calendar
- banner: Print a text in large letters made of the character #
- figlet: Print a text in large letters, in a specific font
- toilet: Print a text in large colorful letters, in a specific font
- lolcat: Print a text in rainbow coloring
- jp2a: Convert a JPG image into ASCII art
- magick: ImageMagick, a versatile tool to edit, transform, and convert images
- fortune: Print a random aphorism, like those found in fortune cookies
- sensors: Print sensor chips information (e.g. temperature)
- beep: Produce a beep from the machine's speakers
- speaker-test: Speaker test tone generator for the ALSA (Advanced Linux Sound Architecture) framework
- on_ac_power: Return 0 (true) if machine is connected to AC power, 1 (false) if on battery. Useful for laptops
- ipcalc: IP addresses calculator
- grepcidr: IP addresses filter against CIDR specifications
- pwgen: Random password generator
- pwqgen: Random password generator with controllable quality
- uuidgen: UUID generator (random or time-based)
- haveged: Random number generator using the HAVEGE (Hardware Volatile Entropy Gathering and Expansion) algorithm. Can be run as a daemon to automatically replenish /dev/random whenever the supply of random bits in the random device gets too low
- aspell: Spell checker
- cloc: Count lines of source code
- nnn: Terminal file manager
- goaccess: Real-time webserver log analyzer with ncurses UI
- gnome-terminal: GNOME shell terminal GUI
- gnome-tweaks: GNOME Tweak Tool
- conky: Highly configurable system monitor widget GUI with integration for audio player, email, and news
- gkrellm: System monitor widget GUI
**Compilers**

- **cc, gcc**: C compiler
- **make**: Utility for automatic compiling, re-compiling, and installation of multi-file programs. It determines automatically which parts of a multi-file program need recompiling. The relationships about these parts, and the commands that must be used to update them, are described in a makefile (`.Makefile` by default).
- **shc**: Shell script compiler, used to prevent a shell script from inspection or modification. It encrypts a shell script, generates C source code, and compiles the C code into a stripped binary executable file.
## Locale environment variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANG</td>
<td>Language, stored in /etc/default/locale.</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>When scripting, it is recommended to set LANG=C because this specifies the minimal locale environment for C translation, and guarantees a standard collation and formats for the execution of scripts.</td>
</tr>
<tr>
<td>LC_CTYPE</td>
<td>Character classification and case conversion</td>
</tr>
<tr>
<td>LC_NUMERIC</td>
<td>Non-monetary numeric formats</td>
</tr>
<tr>
<td>LC_TIME</td>
<td>Date and time formats</td>
</tr>
<tr>
<td>LC_COLLATE</td>
<td>Alphabetical order</td>
</tr>
<tr>
<td>LC_MONETARY</td>
<td>Monetary formats</td>
</tr>
<tr>
<td>LC_MESSAGES</td>
<td>Language and encoding of system messages and user input</td>
</tr>
<tr>
<td>LC_PAPER</td>
<td>Paper size</td>
</tr>
<tr>
<td>LC_NAME</td>
<td>Personal name formats</td>
</tr>
<tr>
<td>LC_ADDRESS</td>
<td>Geographic address formats</td>
</tr>
<tr>
<td>LC_TELEPHONE</td>
<td>Telephone number formats</td>
</tr>
<tr>
<td>LC_MEASUREMENT</td>
<td>Measurement units (metric or others)</td>
</tr>
<tr>
<td>LC_IDENTIFICATION</td>
<td>Metadata about locale</td>
</tr>
<tr>
<td>LC_ALL</td>
<td>Special variable overriding all others</td>
</tr>
</tbody>
</table>

The values of these locale environment variables are in the format `language_territory.encoding` e.g. `en_US.UTF-8`. The list of supported locales is stored in `/usr/share/i18n/SUPPORTED`.

**locale**

Show locale environment variables

**locale-gen it_IT.UTF-8**

Generate a locale (in this case IT) by compiling a list of locale definition files

**apt-get install manpages-it language-pack-it (Debian)**

Install a different locale (in this case IT); this affects system messages and manpages

**iconv -f ISO-8859-10 filein -t UTF-8 > fileout**

Convert a text file from a codeset to another

ISO/IEC-8859 is a standard for 8-bit encoding of printable characters. The first 256 characters in ISO/IEC-8859-1 (Latin-1) are identical to those in Unicode. UTF-8 encoding can represent every character in the Unicode set, and is the de facto standard for text containing characters with diacritics (which do not fit in the ASCII 7-bit set). It was designed for backward compatibility with ASCII. UTF-8 encodes a Unicode character into 8, 16, 24, or 32 bits, whatever necessary; a UTF-8 file containing only ASCII characters is identical to an ASCII file.
**System time**

- `date`
  - `date -d "9999 days ago"`
  - `date -d "1970/01/01 + 4242"`
  - `date +%F %H:%M:%S`
  - `date +%s`
- `timedatectl`
  - `timedatectl set-time 2021-01-04`
  - `timedatectl set-time 23:30`
  - `timedatectl list-timezones`
- `zdump GMT`
- `tzselect`
- `tzconfig`
  - `dpkg-reconfigure tzdata` **(Debian)**
  - `timedatectl set-timezone tz` **(Red Hat)**

**Timezone**

- `/etc/timezone` **(Debian)**
- `/etc/localtime` **(Red Hat)**

**Hardware Clock**

- `hwclock --show`
  - `hwclock -r`
  - `hwclock --hctosys`
  - `hwclock -s`
  - `hwclock --systohc`
  - `hwclock --w`
  - `hwclock --utc`
  - `hwclock --localtime`
The Network Time Protocol is used to keep a machine’s clock in sync with Internet timeservers. It uses UDP port 123.

**ntpd**
- NTP daemon

**ntpd** -q
- Synchronize the time once and quit

**ntpd** -g
- Force NTP to start even if clock is off by more than the panic threshold (1000 secs)

**ntpd** -nqg
- Start NTP as a non-daemon, force synchronization of the clock, and quit.
  - The NTP daemon must not be running when this command is launched

**ntpdc**
- Query and modify the state of the NTP daemon

**ntpq** -p **timeserver**
- Print the list of peers for the timeserver

**ntptrace**
- Trace from where a NTP server gets its time, following the chain of NTP servers back to the primary time source

**ntpddate** **timeserver**
- Synchronizes the clock with the specified timeserver

**ntpddate** -b **timeserver**
- Brutally set the clock, without waiting for it to adjust slowly

**ntpddate** -q **timeserver**
- Query the timeserver without setting the clock

The **ntpddate** command is deprecated; to synchronize the clock, **ntpd** should be used instead.

**chronyd**
- Daemon for chrony, a versatile NTP client/server

**chronyc**
- Command line interface for the chrony daemon
syslogd (Ubuntu 14) Daemon logging events from user processes
rsyslogd Daemon logging events from kernel processes

```
/etc/syslog.conf
# facility.level action
*.*.info;mail.none;authpriv.none /var/log/messages
authpriv.* /var/log/secure
mail.* /var/log/maillog
*.alert
*.emerg
local5.* @10.7.7
local7.* /var/log/boot.log
```

<table>
<thead>
<tr>
<th>Facility</th>
<th>Level</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth or security†</td>
<td>emerg or panic† (highest)</td>
<td>file</td>
</tr>
<tr>
<td>cron</td>
<td>alert</td>
<td>@host</td>
</tr>
<tr>
<td>daemon</td>
<td>crit</td>
<td>user1,user2,user3</td>
</tr>
<tr>
<td>kern</td>
<td>err or error†</td>
<td>*</td>
</tr>
<tr>
<td>lpr</td>
<td>warning or warn†</td>
<td></td>
</tr>
<tr>
<td>mail</td>
<td>notice</td>
<td></td>
</tr>
<tr>
<td>mark (for syslog internal use)</td>
<td>info</td>
<td></td>
</tr>
<tr>
<td>news</td>
<td>debug</td>
<td></td>
</tr>
<tr>
<td>syslog</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>user</td>
<td>(lowest)</td>
<td></td>
</tr>
<tr>
<td>uucp</td>
<td>(facility disabled)</td>
<td></td>
</tr>
<tr>
<td>local0 ... local7 (custom)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† = deprecated

Facilities and levels are listed in the manpage `man 3 syslog`.

```
logger -p auth.info "Message"
Send a message to syslog with facility "auth" and priority "info"
```

logrotate
Rotate logs. It gzips, renames, and eventually deletes old logfiles according to the configuration files `/etc/logrotate.conf` and `/etc/logrotate.d/*`. It is usually scheduled as a daily cron job

```
/var/log/messages
Global system logfile
/var/log/dmesg
Kernel ring buffer information
/var/log/kern.log
Kernel log
/var/log/boot.log
Information logged during boot
```
Mailbox formats

<table>
<thead>
<tr>
<th>Mailbox</th>
<th>Description</th>
<th>File Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>mbox</td>
<td>Each mail folder is a single file, storing multiple email messages. Advantages: universally supported; fast search inside a mail folder. Disadvantages: issues with file locking; possible mailbox corruption.</td>
<td>$HOME/Mail/folder</td>
</tr>
<tr>
<td>Maildir</td>
<td>Each mail folder is a directory, and contains the subdirectories /cur, /new, and /tmp. Each email message is stored in its own file with a unique filename ID. The process that delivers an email message writes it to a file in the tmp/ directory, and then moves it to new/. The moving is commonly done by hard linking the file to new/ and then unlinking the file from tmp/, which guarantees that a MUA will not see a partially written message as it never looks in tmp/. When the MUA finds mail messages in new/ it moves them to cur/. Advantages: fast location/retrieval/deletion of a specific mail message; no file locking needed; can be used with NFS. Disadvantages: some filesystems may not efficiently handle a large number of small files; searching text inside all mail messages is slower.</td>
<td>$HOME/Mail/folder/</td>
</tr>
</tbody>
</table>
### SMTP commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELO xyz.example.com</td>
<td>ESMTM Postfix (server)</td>
</tr>
<tr>
<td>HELO xyz.example.org</td>
<td>Initiate the conversation and identify client host to server</td>
</tr>
<tr>
<td>MAIL FROM: <a href="mailto:alice@linux.org">alice@linux.org</a></td>
<td>Specify mail sender</td>
</tr>
<tr>
<td>RCPT TO <a href="mailto:bob@foobie.com">bob@foobie.com</a></td>
<td>Specify mail recipient</td>
</tr>
<tr>
<td>DATA</td>
<td>Specify data to send. Ended with a dot on a single line</td>
</tr>
<tr>
<td>QUIT</td>
<td>Disconnect</td>
</tr>
<tr>
<td>HELP</td>
<td>List all available commands</td>
</tr>
<tr>
<td>RSET</td>
<td>Empty command</td>
</tr>
<tr>
<td>VRFY <a href="mailto:alice@linux.org">alice@linux.org</a></td>
<td>Verify the existence of an email address (this command should not be implemented, for security reasons)</td>
</tr>
<tr>
<td>EXPN <a href="mailto:list@linux.org">list@linux.org</a></td>
<td>Tell the actual delivery address of aliases and mailing lists</td>
</tr>
</tbody>
</table>

### SMTP response codes

<table>
<thead>
<tr>
<th>First digit</th>
<th>Second digit</th>
<th>Third digit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>211</td>
<td>Command accepted, but not processed until client sends confirmation</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>214</td>
<td>Command successfully completed</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>220</td>
<td>Command accepted, but not processed until client sends more information</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>221</td>
<td>Command failed due to temporary errors</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>250</td>
<td>Command failed due to permanent errors</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>211</td>
<td>Syntax error or command not implemented</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>251</td>
<td>Informative response in reply to a request for information</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>251</td>
<td>Connection response in reply to a data transmission</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>251</td>
<td>Status response in reply to a mail transfer operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Specifies further the response</td>
</tr>
<tr>
<td>211</td>
<td>211</td>
<td>211</td>
<td>System status or help reply</td>
</tr>
<tr>
<td>214</td>
<td>214</td>
<td>214</td>
<td>Help message</td>
</tr>
<tr>
<td>220</td>
<td>220</td>
<td>220</td>
<td>The server is ready</td>
</tr>
<tr>
<td>221</td>
<td>221</td>
<td>221</td>
<td>The server is ending the conversation</td>
</tr>
<tr>
<td>250</td>
<td>250</td>
<td>250</td>
<td>The requested action was completed</td>
</tr>
<tr>
<td>251</td>
<td>251</td>
<td>251</td>
<td>The specified user is not local, but the server will forward the mail message</td>
</tr>
<tr>
<td>354</td>
<td>354</td>
<td>354</td>
<td>The server will be shut down, try again later</td>
</tr>
<tr>
<td>421</td>
<td>421</td>
<td>421</td>
<td>The mailbox that you are trying to reach is busy, try again later</td>
</tr>
<tr>
<td>421</td>
<td>450</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>451</td>
<td>421</td>
<td>The requested action was not done because some error occurred in the mail server</td>
</tr>
<tr>
<td>421</td>
<td>452</td>
<td>421</td>
<td>The mailbox that you are trying to reach has run out of space, try again later</td>
</tr>
<tr>
<td>421</td>
<td>500</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>501</td>
<td>421</td>
<td>The mailbox that you are trying to reach has run out of space, try again later</td>
</tr>
<tr>
<td>421</td>
<td>502</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>503</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>504</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>550</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>551</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>552</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>553</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
<tr>
<td>421</td>
<td>554</td>
<td>421</td>
<td>The mailbox that you are trying to reach cannot be found or you do not have access rights</td>
</tr>
</tbody>
</table>
Sendmail is an MTA distributed as a monolithic binary file. Previous versions used to run SUID root, which caused many security problems; recent versions run SGID smmsp, the group that has write access on the mail queue. Sendmail uses smrsh, a restricted shell, to run some external programs.

Configuration files (must not be edited by hand):

<table>
<thead>
<tr>
<th>/etc/mail/</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>submit.cf</td>
<td>Sendmail local mail transfer configuration file</td>
</tr>
<tr>
<td>sendmail.cf</td>
<td>Sendmail MTA configuration file</td>
</tr>
</tbody>
</table>

m4 /etc/mail/submit.mc > /etc/mail/submit.cf  
Generate a .cf configuration file from an editable .mc text file

Database files (must not be edited by hand):

<table>
<thead>
<tr>
<th>/etc/mail/</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>access.db</td>
<td>Access control file to allow or deny access to systems or users</td>
</tr>
<tr>
<td>local-host-names.db</td>
<td>List of domains that must be considered as local accounts</td>
</tr>
<tr>
<td>virtusertable.db</td>
<td>Map for local accounts, used to distribute incoming email</td>
</tr>
<tr>
<td>mailertable.db</td>
<td>Routing table, used to dispatch emails from remote systems</td>
</tr>
<tr>
<td>domaintable.db</td>
<td>Domain table, used for transitions from an old domain to a new one</td>
</tr>
<tr>
<td>genericsdb.db</td>
<td>Map for local accounts, used to specify a different sender for outgoing mail</td>
</tr>
<tr>
<td>genericsdomain.db</td>
<td>Local FQDN</td>
</tr>
</tbody>
</table>

makemap hash /etc/mail/access.db < /etc/mail/access  
Generate a .db database file from an editable text file

Temporary mailqueue files (where nnn is the Message ID):

<table>
<thead>
<tr>
<th>/var/spool/mqueue/</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfnnn</td>
<td>Mail body</td>
</tr>
<tr>
<td>qfnnn</td>
<td>Message envelope with headers and routing information</td>
</tr>
<tr>
<td>Qfnnn</td>
<td>Message envelope if abandoned</td>
</tr>
<tr>
<td>hfnnn</td>
<td>Message envelope if held / quarantined by a milter (mail filter)</td>
</tr>
<tr>
<td>tfnnn</td>
<td>Temporary file</td>
</tr>
<tr>
<td>ifnnn</td>
<td>Lock file</td>
</tr>
<tr>
<td>nfnnn</td>
<td>Backup file</td>
</tr>
<tr>
<td>xfnnn</td>
<td>Transcript of delivery attempts</td>
</tr>
</tbody>
</table>

newaliases  
sendmail -bi  
mailq  
sendmail -bp  
sendmail -bt  
sendmail -q  
hoststat  
purgestat  
mailstats  
praliases  
Update the aliases database. Must be run after any change to /etc/aliases  
Examine the mail queue  
Run Sendmail in test mode  
Force a queue run  
Print statistics about remote hosts usage  
Clear statistics about remote host usage  
Print statistics about the mailserver  
Display email aliases
Exim is a free MTA, distributed under open source GPL license.

/etc/exim.conf
/usr/local/etc/exim/configure (FreeBSD)

Exim4 configuration file

```
exim4 -bp
exim4 -M messageID
exim4 -Mrm messageID
exim4 -Mvb messageID
exim4 -Mvh messageID
exim4 -qf domain
exim4 -Rff domain
exim4 -bV

exinext
exigrep
exicyclog
```

Examine the mail queue
Attempt delivery of message
Remove a message from the mail queue
See the headers of a message in the mail queue
See the body of a message in the mail queue
See a message in the mail queue
Force a queue run of all queued messages for a domain
Attempt delivery of all queued messages for a domain
Show version and other info
Give the times of the next queue run
Search through Exim logfiles
Rotate Exim logfiles
Postfix is a fast, secure, easy to configure, open source MTA intended as a replacement for Sendmail. It is implemented as a set of small helper daemons, most of which run in a chroot jail with low privileges. The main ones are:

- **master**: Postfix master daemon, always running; starts the other daemons when necessary
- **nqmgr**: Queue manager for incoming and outgoing mail, always running
- **smtpd**: SMTP daemon for incoming mail
- **smtp**: SMTP daemon for outgoing mail
- **bounce**: Manager of bounce messages
- **cleanup**: Daemon that verifies the syntax of outgoing messages before they are handed to the queue manager
- **local**: Daemon that handles local mail delivery
- **virtual**: Daemon that handles mail delivery to virtual users

### Queue Directories

```
/var/spool/postfix/
```

- **incoming**: Incoming queue. All new mail entering the Postfix queue is written here by the cleanup daemon. Under normal conditions this queue is nearly empty.
- **active**: Active queue. Contains messages ready to be sent. The queue manager places messages here from the incoming queue as soon as they are available.
- **deferred**: Deferred queue. A message is placed here when all its deliverable recipients are delivered, and delivery failed for some recipients for a transient reason. The queue manager scans this queue periodically and puts some messages back into the active queue to retry sending.
- **bounce**: Message delivery status report about why mail is bounced (non-delivered mail)
- **defer**: Message delivery status report about why mail is delayed (non-delivered mail)
- **trace**: Message delivery status report (delivered mail)

### Command Summary

- **postfix reload**: Reload configuration
- **postconf -e 'mydomain = example.org'**: Edit a setting in the Postfix configuration
- **postconf -l**: List supported mailbox lock methods
- **postconf -m**: List supported database types
- **postconf -v**: Increase logfile verbosity
- **postmap dbtype:textfile**: Manage Postfix lookup tables, creating a hashed map file of database type `dbtype` from `textfile`
- **postmap hash:/etc/postfix/transport**: Regenerate the transport database
- **postalias**: Convert `/etc/aliases` into the `aliases` database file `/etc/aliases.db`
- **postsuper**: Operate on the mail queue
- **postqueue**: Unprivileged mail queue manager
## Postfix - configuration

### /etc/postfix/main.cf  
**Postfix main configuration file**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mydomain = example.org</td>
<td>This system’s domain</td>
</tr>
<tr>
<td>myorigin = $mydomain</td>
<td>Domain from which all sent mail will appear to originate</td>
</tr>
<tr>
<td>myhostname = foobar.$mydomain</td>
<td>This system’s hostname</td>
</tr>
<tr>
<td>inet_interfaces = all</td>
<td>Network interface addresses that this system receives mail on. Value can also be localhost, all, or loopback-only</td>
</tr>
<tr>
<td>proxy_interfaces = 1.2.3.4</td>
<td>Network interface addresses that this system receives mail on by means of a proxy or NAT unit</td>
</tr>
<tr>
<td>mynetworks = 10.3.3.0/24 !10.3.3.66</td>
<td>Networks the SMTP clients are allowed to connect from</td>
</tr>
<tr>
<td>mydestination = $mydomain, localhost, $mydomain, example.com, hash:/etc/postfix/otherdomains</td>
<td>Domains for which Postfix will accept received mail. Value can also be a lookup database file e.g. a hashed map</td>
</tr>
<tr>
<td>relayhost = 10.6.6.6</td>
<td>Relay host to which Postfix should send all mail for delivery, instead of consulting DNS MX records</td>
</tr>
<tr>
<td>relay_domains = $mydestination</td>
<td>Sources and destinations for which mail will be relayed. Can be empty if Postfix is not intended to be a mail relay</td>
</tr>
<tr>
<td>virtual_alias_domains = virtualex.org</td>
<td>Set up Postfix to handle mail for virtual domains too. The /etc/postfix/virtual file is a hashed map, each line of the file containing the virtual domain email address and the destination real domain email address: <a href="mailto:jdoe@virtualex.org">jdoe@virtualex.org</a>    <a href="mailto:john.doe@example.org">john.doe@example.org</a>  <a href="mailto:ksmith@virtualex.org">ksmith@virtualex.org</a>    kim.smith  @virtualex.org    root  The @virtualex.org in the last line is a catch-all specifying that all other email messages to the virtual domain are delivered to the root user on the real domain</td>
</tr>
<tr>
<td>mailbox_command = /usr/bin/procmail</td>
<td>Use Procmail as MDA</td>
</tr>
</tbody>
</table>

A line beginning with whitespace or tab is a continuation of the previous line. A # not placed at the beginning of a line is not a comment delimiter.

### /etc/postfix/master.cf  
**Postfix master daemon configuration file**

<table>
<thead>
<tr>
<th>Service</th>
<th>Name of the service</th>
<th>Transport mechanism used by the service</th>
<th>Whether the service is accessible only by Postfix daemons and not by the whole system. Default is yes</th>
<th>Whether the service is unprivileged i.e. not running as root. Default is yes</th>
<th>Whether the service is chrooted. Default is yes</th>
<th>How often the service needs to be woken up by the master daemon. Default is never</th>
<th>Max number of simultaneous processes providing the service. Default is 50</th>
<th>Command used to start the service</th>
</tr>
</thead>
<tbody>
<tr>
<td>smtp</td>
<td>smtpd</td>
<td>inet</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>pickup</td>
<td>pickup</td>
<td>fifo n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>60</td>
<td>1</td>
<td>n</td>
</tr>
<tr>
<td>cleanup</td>
<td>cleanup</td>
<td>unix n</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>qmgr</td>
<td>qmgr</td>
<td>fifo n</td>
<td>-</td>
<td>-</td>
<td>300</td>
<td>1</td>
<td>qmgr</td>
<td>-</td>
</tr>
<tr>
<td>rewrite</td>
<td>trivial-rewrite</td>
<td>unix -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bounce</td>
<td>bounce</td>
<td>unix -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>bounce</td>
</tr>
<tr>
<td>defer</td>
<td>bounce</td>
<td>unix -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>bounce</td>
</tr>
<tr>
<td>flush</td>
<td>flush</td>
<td>unix n</td>
<td>-</td>
<td>-</td>
<td>1000?</td>
<td>0</td>
<td>flush</td>
<td>-</td>
</tr>
<tr>
<td>smtp</td>
<td>smtpd</td>
<td>unix -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>smtpd</td>
</tr>
<tr>
<td>showq</td>
<td>showq</td>
<td>unix n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>showq</td>
</tr>
<tr>
<td>error</td>
<td>error</td>
<td>unix -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>local</td>
<td>local</td>
<td>unix -</td>
<td>n</td>
<td>n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>local</td>
</tr>
<tr>
<td>virtual</td>
<td>virtual</td>
<td>unix -</td>
<td>n</td>
<td>n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>virtual</td>
</tr>
<tr>
<td>lmtp</td>
<td>lmtp</td>
<td>unix -</td>
<td>n</td>
<td>n</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>lmtp</td>
</tr>
</tbody>
</table>

The - indicates that an option is set to its default value.
Procmail is a regex-based MDA whose main purpose is to preprocess and sort incoming email messages. It is able to work both with the standard mbox format and the Maildir format.

To have all email processed by Procmail, the `.forward` file may be edited to contain:

```
|exec /usr/local/bin/procmail || exit 75
```

Common parameters, nonspecific to Procmail:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td><code>$HOME/bin:/usr/bin:/bin:/usr/sbin:/sbin</code></td>
</tr>
<tr>
<td>MAILDIR</td>
<td><code>$HOME/Mail</code></td>
</tr>
<tr>
<td>DEFAULT</td>
<td><code>$MAILDIR/Inbox</code></td>
</tr>
<tr>
<td>LOGFILE</td>
<td><code>$HOME/.procmaillog</code></td>
</tr>
</tbody>
</table>

Flag: match headers (default) and use file locking (highly recommended when writing to a file or a mailbox in mbox format)
Condition: match the header specifying the sender address
Destination: default mailfolder

Flag: file locking not necessary because using Maildir format
Conditions: match sender address and subject headers
Destination: specified mailfolder, in Maildir format

Flag: file locking not necessary because blackholing to `/dev/null`
Condition: match SpamAssassin's specific header
Destination: delete the message

Flag: match body of message instead of headers

Flag: match either headers or body of message

Condition: match messages larger than 256 Kb
Destination: pipe message through the specified program

Flags: use the pipe as a filter (modifying the message), and have Procmail wait that the filter finished processing the message

Flag: copy the message and proceed with next recipe
Destination: forward to specified email address, and (this is ordered by the next recipe) save in the specified mailfolder
The Courier MTA provides modules for ESMTP, IMAP, POP3, webmail, and mailing list services in a single framework. To use Courier, it is necessary first to launch the `courier-authlib` service, then launch the desired mail service e.g. `courier-imap` for the IMAP service.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lib/courier-imap/etc/</td>
<td>Courier IMAP daemon configuration</td>
</tr>
<tr>
<td>/etc/courier/</td>
<td>Courier IMAPS daemon configuration</td>
</tr>
<tr>
<td>/usr/lib/courier-imap/share/</td>
<td>Courier POP3 daemon configuration</td>
</tr>
<tr>
<td>/usr/lib/courier-imap/share/</td>
<td>Directory for public and private keys</td>
</tr>
<tr>
<td>mkimapdcert</td>
<td>Generate a certificate for the IMAPS service</td>
</tr>
<tr>
<td>mkpop3dcert</td>
<td>Generate a certificate for the POP3 service</td>
</tr>
<tr>
<td>makealiases</td>
<td>Create system aliases in <code>/usr/lib/courier/etc/aliases.dat</code>, which is made by processing a <code>/usr/lib/courier/etc/aliases/system</code> text file:</td>
</tr>
<tr>
<td>root</td>
<td>: postmaster</td>
</tr>
<tr>
<td>mailer-daemon</td>
<td>: postmaster</td>
</tr>
<tr>
<td>MAILER-DAEMON</td>
<td>: postmaster</td>
</tr>
<tr>
<td>uucp</td>
<td>: postmaster</td>
</tr>
<tr>
<td>postmaster</td>
<td>: admin</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lib/courier-imap/etc/pop3d</td>
<td>Courier POP configuration file</td>
</tr>
<tr>
<td>ADDRESS=0</td>
<td>Address on which to listen. 0 means all addresses</td>
</tr>
<tr>
<td>PORT=127.0.0.1.900,192.168.0.1.900</td>
<td>Port number on which connections are accepted. In this case, accept connections on port 900 on IP addresses 127.0.0.1 and 192.168.0.1</td>
</tr>
<tr>
<td>POP3AUTH=&quot;LOGIN CRAM-MD5 CRAM-SHA1&quot;</td>
<td>POP authentication advertising SASL (Simple Authentication and Security Layer) capability, with CRAM-MD5 and CRAM-SHA1</td>
</tr>
<tr>
<td>POP3AUTH_TLS=&quot;LOGIN PLAIN&quot;</td>
<td>Also advertise SASL PLAIN if SSL is enabled</td>
</tr>
<tr>
<td>MAXDAEMONS=40</td>
<td>Maximum number of POP3 servers started</td>
</tr>
<tr>
<td>MAXPERIP=4</td>
<td>Maximum number of connections to accept from the same IP address</td>
</tr>
<tr>
<td>PIDFILE=/var/run/courier/pop3d.pid</td>
<td>PID file</td>
</tr>
<tr>
<td>TCPDOPTS=&quot;-nodnslookup -noidentlookup&quot;</td>
<td>Miscellaneous <code>couriertcpd</code> options. Should not be changed</td>
</tr>
<tr>
<td>LOGGEROPTS=&quot;-name=pop3d&quot;</td>
<td>Options for <code>courierlogger</code></td>
</tr>
<tr>
<td>POP3_PROXY=0</td>
<td>Enable or disable proxying</td>
</tr>
<tr>
<td>PROXY_HOSTNAME=myproxy</td>
<td>Override value from <code>gethostname()</code> when checking if a proxy connection is required</td>
</tr>
<tr>
<td>DEFDOMAIN=&quot;@example.com&quot;</td>
<td>Optional default domain. If the username does not contain the first character of <code>DEFDOMAIN</code>, then it is appended to the username. If <code>DEFDOMAIN</code> and <code>DOMAINSEP</code> are both set, then <code>DEFDOMAIN</code> is appended only if the username does not contain any character from <code>DOMAINSEP</code></td>
</tr>
<tr>
<td>POP3DSTART=YES</td>
<td>Flag intended to be read by the system startup script</td>
</tr>
<tr>
<td>MAILDIRPATH=Maildir</td>
<td>Maildir directory</td>
</tr>
<tr>
<td>ADDRESS=0</td>
<td>Address on which to listen. 0 means all addresses</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>PORT=127.0.0.1.900,192.168.0.1.900</td>
<td>Port number on which connections are accepted. In this case, accept connections on port 900 on IP addresses 127.0.0.1 and 192.168.0.1</td>
</tr>
<tr>
<td>AUTHSERVICE143=imap</td>
<td>Authenticate using a different service parameter depending on the connection's port. This only works with authentication modules that use the service parameter, such as PAM</td>
</tr>
<tr>
<td>MAXGAMMONS=40</td>
<td>Maximum number of IMAP servers started</td>
</tr>
<tr>
<td>MAXPERIP=20</td>
<td>Maximum number of connections to accept from the same IP address</td>
</tr>
<tr>
<td>PIDFILE=/var/run/courier/imap.pid</td>
<td>PID file for courier tcpd</td>
</tr>
<tr>
<td>TCPDOPTS=&quot;-nodnslookup -noidentlookup&quot;</td>
<td>Miscellaneous courier tcpd options. Should not be changed</td>
</tr>
<tr>
<td>LOGGEROPTS=&quot;-name=imapd&quot;</td>
<td>Options for courier logger</td>
</tr>
<tr>
<td>DEFDOMAIN=&quot;@example.com&quot;</td>
<td>Optional default domain. If the username does not contain the first character of DEFDOMAIN, then it is appended to the username. If DEFDOMAIN and DOMAINSEP are both set, then DEFDOMAIN is appended only if the username does not contain any character from DOMAINSEP</td>
</tr>
<tr>
<td>IMAP_CAPABILITY=&quot;IMAP4rev1 UIDPLUS \ CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT \ THREAD=REFERENCES SORT QUOTA IDLE&quot;</td>
<td>Specifies what most of the response should be to the CAPABILITY command</td>
</tr>
<tr>
<td>IMAP_KEYWORSD=1</td>
<td>Enable or disable custom IMAP keywords. Possible values are: 0 disable keywords 1 enable keywords 2 enable keywords with a slower algorithm</td>
</tr>
<tr>
<td>IMAP_ACL=1</td>
<td>Enable or disable IMAP ACL extension</td>
</tr>
<tr>
<td>SMAP_CALLABILITY=SMAP1</td>
<td>Enable the experimental Simple Mail Access Protocol extensions</td>
</tr>
<tr>
<td>IMAP_PROXY=0</td>
<td>Enable or disable proxying</td>
</tr>
<tr>
<td>IMAP_PROXY_FOREIGN=0</td>
<td>Proxying to non-Courier servers. Resends the CAPABILITY command after logging in to remote server. May not work with all IMAP clients</td>
</tr>
<tr>
<td>IMAP_IDLE_TIMEOUT=60</td>
<td>How often, in seconds, the server should poll for changes to the folder while in IDLE mode</td>
</tr>
<tr>
<td>IMAP_CHECK_ALL FOLDERS=0</td>
<td>Enable or disable server check for mail in every folder</td>
</tr>
<tr>
<td>IMAP_UMASK=022</td>
<td>Set the umask of the server process. This value is passed to the umask command. Mostly useful for shared folders, where file permissions of the messages may be important</td>
</tr>
<tr>
<td>IMAP_ULIMITD=131072</td>
<td>Set the upper limit of the size of the data segment of the server process, in Kb. This value is passed to the ulimit -d command. Used as an additional safety check to stop potential DoS attacks that exploit memory leaks to exhaust all the available RAM on the server</td>
</tr>
<tr>
<td>IMAP_USELOCKS=1</td>
<td>Enable or disable dot-locking to support concurrent multiple access to the same folder. Strongly recommended when using shared folders</td>
</tr>
<tr>
<td>IMAP_SHAREDINDEXFILE=&quot;/etc/courier/shared/index&quot;</td>
<td>Index of all accessible folders. This setting should normally not be changed</td>
</tr>
<tr>
<td>IMAP_TRASHFOLDERNAME=Trash</td>
<td>Trash folder</td>
</tr>
<tr>
<td>IMAP_EMPTYTRASH=Trash:7,Sent:30</td>
<td>Purge folders i.e. delete all messages from the specified folders after the specified number of days</td>
</tr>
<tr>
<td>IMAP_MOVE_EXPUNGE_TO TRASH=0</td>
<td>Enable or disable moving expunged messages to the trash folder (instead of directly deleting them)</td>
</tr>
<tr>
<td>HEADERFROM=X-IMAP-Sender</td>
<td>Save the return address ($SENDER) in the X-IMAP-Sender mail header. This header is added to the sent message, but not in the copy of the message saved in the folder</td>
</tr>
<tr>
<td>MAILDIRPATH=Maildir</td>
<td>Mail directory</td>
</tr>
</tbody>
</table>
Dovecot is an open source, security-hardened, fast, and efficient IMAP and POP3 server. It implements its own high-performance dbox mailbox format. By default, it uses PAM authentication. The script `mkcert.sh` can be used to create self-signed SSL certificates.

### Dovecot configuration file

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>base_dir = /var/run/dovecot/</code></td>
<td>Base directory where to store runtime data</td>
</tr>
<tr>
<td><code>protocols = imaps pop3s</code></td>
<td>Protocols to serve. If Dovecot should use <code>dovecot-auth</code>, this can be set to <code>none</code></td>
</tr>
<tr>
<td><code>listen = *, [::]</code></td>
<td>Network interfaces on which to accept connections. In this case, listen to all IPv4 and IPv6 interfaces</td>
</tr>
<tr>
<td><code>disable_plaintext_auth = yes</code></td>
<td>If yes, disable LOGIN command and all other plaintext authentications unless SSL/TLS is used (LOGINDISABLED capability)</td>
</tr>
<tr>
<td><code>shutdown_clients = yes</code></td>
<td>If yes, kill all IMAP and POP3 processes when Dovecot master process shuts down; if no, Dovecot can be upgraded without forcing existing client connections to close</td>
</tr>
<tr>
<td><code>log_path = /dev/stderr</code></td>
<td>Log file to use for error messages, instead of sending them to syslog. In this case, log to stderr</td>
</tr>
<tr>
<td><code>info_log_path = /dev/stderr</code></td>
<td>Log file to use for informational and debug messages. Default value is the same as <code>log_path</code></td>
</tr>
<tr>
<td><code>syslog_facility = mail</code></td>
<td>Syslog facility to use, if logging to syslog</td>
</tr>
<tr>
<td><code>login_dir = /var/run/dovecot/login</code></td>
<td>Directory where the authentication process places authentication UNIX sockets. The login process needs to be able to connect to these sockets</td>
</tr>
<tr>
<td><code>login_chroot = yes</code></td>
<td>Chroot login process to the <code>login_dir</code></td>
</tr>
<tr>
<td><code>login_user = dovecot</code></td>
<td>User for the login process and for access control in the authentication process. This is not the user that will access mail messages</td>
</tr>
<tr>
<td><code>login_process_size = 64</code></td>
<td>Maximum login process size, in Mb</td>
</tr>
<tr>
<td><code>login_process_per_connection = yes</code></td>
<td>If yes, each login is processed in its own process (more secure); if no, each login process processes multiple connections (faster)</td>
</tr>
<tr>
<td><code>login_processes_count = 3</code></td>
<td>Number of login processes to keep for listening for new connections</td>
</tr>
<tr>
<td><code>login_max_processes_count = 128</code></td>
<td>Maximum number of login processes to create</td>
</tr>
<tr>
<td><code>login_max_connections = 256</code></td>
<td>Maximum number of connections allowed per each login process. This setting is used only if <code>login_process_per_connection = no</code>; once the limit is reached, the process notifies master so that it can create a new login process</td>
</tr>
<tr>
<td><code>login_greeting = Dovecot ready.</code></td>
<td>Greeting message for clients</td>
</tr>
<tr>
<td><code>login_trusted_networks = </code></td>
<td>Trusted network ranges (usually IMAP proxy servers). Connections from these IP addresses are allowed to override their IP addresses and ports, for logging and authentication checks. <code>disable_plaintext_auth</code> is also ignored for these networks</td>
</tr>
<tr>
<td><code>mbox_read_locks = fcntl</code></td>
<td>Locking methods to use for locking mailboxes in mbox format. Possible values are:</td>
</tr>
<tr>
<td><code>mbox_write_locks = dotlock fcntl</code></td>
<td><code>dotlock</code> Create mailbox.lock file; oldest and NSF-safe method</td>
</tr>
<tr>
<td></td>
<td><code>dotlock_try</code> Same as <code>dotlock</code>, but skip if failing</td>
</tr>
<tr>
<td></td>
<td><code>fcntl</code> Recommended; works with NFS too if <code>lockd</code> is used</td>
</tr>
<tr>
<td></td>
<td><code>flock</code> May not exist in all systems; doesn’t work with NFS</td>
</tr>
<tr>
<td></td>
<td><code>lockf</code> May not exist in all systems; doesn’t work with NFS</td>
</tr>
<tr>
<td><code>maildir_stat_dirs = no</code></td>
<td>Option for mailboxes in Maildir format. If no (default), the LIST command returns all entries in the mail directory beginning with a dot; if yes, returns only entries which are directories</td>
</tr>
<tr>
<td><code>dbox_rotate_size = 2048</code></td>
<td>Maximum and minimum file size, in Kb, of a mailbox in dbox format until it is rotated</td>
</tr>
<tr>
<td><code>dbox_rotate_min_size = 16</code></td>
<td>Include configuration file</td>
</tr>
<tr>
<td><code>!include /etc/dovecot/conf.d/*.conf</code></td>
<td>Include configuration file, and do not report an error if file is not found</td>
</tr>
<tr>
<td><code>!include_try /etc/dovecot/extra.conf</code></td>
<td>Include optional configuration file, and do not report an error if file is not found</td>
</tr>
</tbody>
</table>
### Dovecot - mailbox configuration

<table>
<thead>
<tr>
<th>/etc/dovecot.conf</th>
<th>Dovecot configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail_location = \</td>
<td>Mailbox location, in mbox or Maildir format. Variables:</td>
</tr>
<tr>
<td>\mailbox:/mail:INBOX=/var/spool/mail/%u</td>
<td>%u username</td>
</tr>
<tr>
<td>or \</td>
<td>%n user part in user@domain, same as %u if there is no domain</td>
</tr>
<tr>
<td>mail_location = maildir:~/Maildir \</td>
<td>%d domain part in user@domain, empty if there is no domain</td>
</tr>
<tr>
<td>\</td>
<td>%h home directory</td>
</tr>
<tr>
<td>namespace shared { \</td>
<td>Definition of a shared namespace, for accessing other users' mailboxes that have been shared.</td>
</tr>
<tr>
<td>\</td>
<td>Private namespaces are for users' personal emails.</td>
</tr>
<tr>
<td>\</td>
<td>Public namespaces are for shared mailboxes managed by root user</td>
</tr>
<tr>
<td>\</td>
<td>Hierarchy separator to use. It should be the same for all namespaces, and depends on the underlying mail storage format</td>
</tr>
<tr>
<td>\</td>
<td>Prefix required to access this namespace; must be different for each.</td>
</tr>
<tr>
<td>\</td>
<td>In this case, mailboxes are visible under shared/user@domain/; the variables %n, %d, and %u are expanded to the destination user</td>
</tr>
<tr>
<td>\</td>
<td>Mailbox location for other users' mailboxes; it is in the same format as mail_location which is also the default for it.</td>
</tr>
<tr>
<td>\</td>
<td>%variable and */ expand to the logged in user's data; %variable expands to the destination user's data</td>
</tr>
<tr>
<td>\</td>
<td>Define whether this namespace contains the INBOX. Note that there can be only one INBOX across all namespaces</td>
</tr>
<tr>
<td>\</td>
<td>Define whether the namespace is hidden i.e. not advertised to clients via NAMESPACE extension</td>
</tr>
<tr>
<td>\</td>
<td>Namespace handles its own subscriptions; if set to no, the parent namespace handles them and Dovecot uses the default namespace for saving subscriptions. If prefix is empty, this should be set to yes</td>
</tr>
<tr>
<td>\</td>
<td>Show the mailboxes under this namespace with LIST command, making the namespace visible for clients that do not support the NAMESPACE extension. In this case, lists child mailboxes but hide the namespace prefix; list the namespace only if there are visible shared mailboxes</td>
</tr>
<tr>
<td>\</td>
<td>}</td>
</tr>
<tr>
<td>mail_uid = 666 \</td>
<td>UID and GID used to access mail messages</td>
</tr>
<tr>
<td>mail_gid = 666 \</td>
<td>Group to enable temporarily for privileged operations. Currently this is used only with INBOX when its initial creation or a dotlocking fails</td>
</tr>
<tr>
<td>mail_privileged_group = mail \</td>
<td>Supplementary groups to with grant access for mail processes. Used typically to set up access to shared mailboxes</td>
</tr>
<tr>
<td>mail_access_groups = tmpmail \</td>
<td>Locking method for index files. Can be fcntl, flock, or dotlock</td>
</tr>
<tr>
<td>lock_method = fcntl \</td>
<td>Valid UID range for users; default is 500 and above. This makes sure that users cannot login as daemons or other system users. Denying root login is hardcoded to Dovecot and cannot be bypassed</td>
</tr>
<tr>
<td>first_valid_uid = 500 \</td>
<td>Valid GID range for users; default is non-root. Users with invalid primary GID are not allowed to login</td>
</tr>
<tr>
<td>last_valid_uid = 0 \</td>
<td>Maximum number of running mail processes. When this limit is reached, new users are not allowed to login</td>
</tr>
<tr>
<td>first_valid_gid = 1 \</td>
<td>Maximum mail process size, in Mb</td>
</tr>
<tr>
<td>last_valid_gid = 0 \</td>
<td>List of directories under which chrooting is allowed for mail processes</td>
</tr>
<tr>
<td>max_mail_processes = 512 \</td>
<td>Default chroot directory for mail processes. Usually not needed as Dovecot does not allow users to access files outside their mail directory</td>
</tr>
<tr>
<td>max_mail_processes = 512 \</td>
<td>Minimum time, in seconds, to wait between mailbox checks. When the IDLE command is running, mailbox is checked periodically for new mails or other changes</td>
</tr>
<tr>
<td>mail_process_size = 256 \</td>
<td>mailbox_idle_check_interval = 30</td>
</tr>
</tbody>
</table>
Dovecot - POP and IMAP configuration

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/dovecot.conf</td>
<td>Dovecot configuration file</td>
</tr>
</tbody>
</table>

```plaintext
protocol pop3 {
    listen = *:110
    login_executable = /usr/libexec/dovecot/pop3-login
    mail_executable = /usr/libexec/dovecot/pop3
    pop3_no_flag_updates = no
    pop3_lock_session = no
    pop3_uidl_format = %08Xu%08Xv
}
```

```plaintext
protocol imap {
    listen = *:143
    ssl_listen = *:993
    login_executable = /usr/libexec/dovecot/imap-login
    mail_executable = /usr/libexec/dovecot/imap
    mail_max_userip_connections = 10
    imap_idle_notify_interval = 120
}
```

SSL/TLS support.
Possible values are yes, no, required

```plaintext
ssl = yes
ssl_cert_file = /etc/ssl/certs/dovecot-cert.pem
ssl_key_file = /etc/ssl/private/dovecot-key.pem
ssl_key_password = p4ssw0rd
ssl_ca_file = /etc/dovecot/cafile.pem
ssl_verify_client_cert = yes
ssl_cipher_list = ALL:!LOW:!SSLv2
verbose_ssl = yes
```

List of trusted SSL certificate authorities.
This file contains CA certificates followed by CRLs

Request client to send a certificate

List of SSL ciphers to use

Show protocol level SSL errors
Dovecot - authentication

/etc/dovecot.conf  Dovecot configuration file

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth_executable</td>
<td>/usr/libexec/dovecot/dovecot-auth</td>
<td>Location of the authentication executable</td>
</tr>
<tr>
<td>auth_process_size</td>
<td>256</td>
<td>Max authentication process size, in Mb</td>
</tr>
<tr>
<td>auth_username_chars</td>
<td>abcde ... VWXY201234567890.-_@</td>
<td>List of allowed characters in the username. If the username entered by the user contains a character not listed in here, the login automatically fails. This is to prevent a user exploiting any potential quote-escaping vulnerabilities with SQL/LDAP databases</td>
</tr>
<tr>
<td>auth_realms</td>
<td></td>
<td>List of realms for SASL authentication mechanisms that need them. If empty, multiple realms are not supported</td>
</tr>
<tr>
<td>auth_default_realm</td>
<td>example.org</td>
<td>Default realm/domain to use if none was specified</td>
</tr>
<tr>
<td>auth_anonymous_username</td>
<td>anonymous</td>
<td>Username to assign to users logging in with ANONYMOUS SASL mechanism</td>
</tr>
<tr>
<td>auth_verbose</td>
<td>no</td>
<td>Defines whether to log unsuccessful authentication attempts and the reasons why they failed</td>
</tr>
<tr>
<td>auth_debug</td>
<td>no</td>
<td>Define whether to enable more verbose logging (e.g. SQL queries) for debugging purposes</td>
</tr>
<tr>
<td>auth_failure_delay</td>
<td>2</td>
<td>Delay before replying to failed authentications, in seconds</td>
</tr>
<tr>
<td>auth default</td>
<td></td>
<td>Accepted authentication mechanisms</td>
</tr>
<tr>
<td></td>
<td>mechanisms = plain login cram-md5</td>
<td>Deny login to the users listed in /etc/dovecot.deny (this file contains one user per line)</td>
</tr>
<tr>
<td></td>
<td>passdb passwd-file {</td>
<td>PAM authentication block.</td>
</tr>
<tr>
<td></td>
<td>args = /etc/dovecot.deny</td>
<td>Enables authentication matching (username and remote IP address) for PAM</td>
</tr>
<tr>
<td></td>
<td>deny = yes</td>
<td>System users e.g. NSS or /etc/passwd</td>
</tr>
<tr>
<td></td>
<td>passdb pam {</td>
<td>Shadow passwords for system users, e.g. NSS or /etc/passwd</td>
</tr>
<tr>
<td></td>
<td>args = cache_key=%u%r dovecot</td>
<td>PAM-like authentication for OpenBSD</td>
</tr>
<tr>
<td></td>
<td>passdb passwd {</td>
<td>SQL database</td>
</tr>
<tr>
<td></td>
<td>blocking = yes</td>
<td>LDAP database</td>
</tr>
<tr>
<td></td>
<td>args =</td>
<td>Export the authentication interface to other programs. Master socket provides access to userdb information, and is typically used to give Dovecot's local delivery agent access to userdb so it can find mailbox locations. The default user/group is the one who started dovecot-auth (i.e. root). The client socket is generally safe to export to everyone. Typical use is to export it to the SMTP server so it can do SMTP AUTH lookups using it</td>
</tr>
<tr>
<td></td>
<td>passdb shadow {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>blocking = yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>args =</td>
<td></td>
</tr>
<tr>
<td></td>
<td>passdb bsdauth {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cache_key = %u</td>
<td></td>
</tr>
<tr>
<td></td>
<td>args =</td>
<td></td>
</tr>
<tr>
<td></td>
<td>passdb sql {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>args = /etc/dovecot/dovecot-sql.conf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>passdb ldap {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>args = /etc/dovecot/dovecot-ldap.conf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>socket listen {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>master {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>path = /var/run/dovecot/auth-master</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mode = 0600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>user =</td>
<td></td>
</tr>
<tr>
<td></td>
<td>group =</td>
<td></td>
</tr>
<tr>
<td></td>
<td>client {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>path = /var/run/dovecot/auth-client</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mode = 0660</td>
<td></td>
</tr>
</tbody>
</table>
FTP (File Transfer Protocol) is a client-server unencrypted protocol for file transfer. Secure alternatives are FTPS (FTP secured with SSL/TLS) and SFTP (SSH File Transfer Protocol). FTP can operate either in active or in passive mode.

**Active mode** (default)
1. Client connects to FTP server on port 21 (control channel) and sends second unprivileged port number
2. Server acknowledges
3. Server connects from port 20 (data channel) to client's second unprivileged port number
4. Client acknowledges

**Passive mode** (more protocol-compliant, because it is the client, not the server, that initiates the second connection)
1. Client connects to FTP server on port 21 and requests passive mode via the PASV command
2. Server acknowledges and sends unprivileged port number via the PORT command
3. Client connects to server's unprivileged port number
4. Server acknowledges

### FTP servers

<table>
<thead>
<tr>
<th>Very Secure FTP</th>
<th>Hardened and high-performance FTP implementation. The vsftpd daemon operates with multiple processes that run as a non-privileged user in a chrooted jail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure-FTP</td>
<td>Free and easy-to-use FTP server</td>
</tr>
<tr>
<td></td>
<td>pure-ftpd</td>
</tr>
<tr>
<td></td>
<td>pure-ftpwho Show clients connected to the Pure-FTP server</td>
</tr>
<tr>
<td></td>
<td>pure-mrtginfo Show connections to the Pure-FTP server as a MRTG graph</td>
</tr>
<tr>
<td></td>
<td>pure-statsdecode Show Pure-FTP log data</td>
</tr>
<tr>
<td></td>
<td>pure-pw Manage Pure-FTP virtual accounts</td>
</tr>
<tr>
<td></td>
<td>pure-pwconvert Convert the system user database to a Pure-FTP virtual accounts database</td>
</tr>
<tr>
<td></td>
<td>pure-quotacheck Manage Pure-FTP quota database</td>
</tr>
<tr>
<td></td>
<td>pure-uploadscript Run a command on the Pure-FTP server to process an uploaded file</td>
</tr>
</tbody>
</table>

### FTP clients

<table>
<thead>
<tr>
<th>ftp</th>
<th>Standard FTP client</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp</td>
<td>Connect to an FTP server</td>
</tr>
<tr>
<td>ftpserver.domain.com</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>lftp</th>
<th>Sophisticated FTP client with support for HTTP and BitTorrent</th>
</tr>
</thead>
<tbody>
<tr>
<td>lftp</td>
<td>Connect to an FTP server and try an anonymous login</td>
</tr>
<tr>
<td>ftpserver.domain.com</td>
<td></td>
</tr>
</tbody>
</table>
/etc/vsftpd/vsftpd.conf  Very Secure FTP server configuration file

- `listen=YES` Run vsftpd in standalone mode (i.e. not via inetd)?
- `local_enable=YES` Allow local system users (i.e. in /etc/passwd) to log in?
- `chroot_local_user=YES` Chroot local users in their home directory?
- `write_enable=YES` Allow FTP commands that write on the filesystem (i.e. STOR, DELE, RNFR, RNTO, MKD, RMD, APPE, and SITE)?
- `anonymous_enable=YES` Allow anonymous logins? If yes, anonymous and ftp are accepted as logins
- `anon_root=/var/ftp/pub` Directory to go after anonymous login
- `anon_upload_enable=YES` Allow anonymous uploads?
- `chown_uploads=YES` Change ownership of anonymously uploaded files?
- `chown_username=ftp` User to whom set ownership of anonymously uploaded files
- `anon_world_readable_only=NO` Allow anonymous users to only download world-readable files?
- `ssl_enable=YES` Enable SSL?
- `force_local_data_ssl=YES` Encrypt local data?
- `force_local_logins_ssl=YES` Force encrypted authentication?
- `allow_anon_ssl=YES` Allow anonymous users to use SSL?
- `ssl_tlsv1=YES` Allowed SSL/TLS versions
- `ssl_tlsv2=NO` Location of certificate file
- `ssl_tlsv3=NO` Location of private key file
- `rsa_cert_file=/etc/pki/tls/certs/vsftpd.pem`
- `rsa_private_key_file=/etc/pki/tls/certs/vsftpd.pem`
In Linux, printers are managed by `cupsd`, the CUPS (Common Unix Printing System) daemon. Printers are administered via a web interface on the URL http://localhost:631.

```
/etc/cups/cupsd.conf
/etc/cups/printers.conf
/etc/printcap
/var/spool/cups/
/var/log/cups/error_log
/var/log/cups/page_log
```

- **CUPS configuration file**
- **Database of available local CUPS printers**
- **Database of printer capabilities, for old printing applications**
- **Printer spooler for data awaiting to be printed**
- **CUPS error log**
- **Information about printed pages**

```
/etc/init.d/cupsys start
```

- **Start the CUPS service**

```
gnome-cups-manager
cupsenable printer0
cupsdisable printer0
cupsaccept printer0
cupsreject -r "Message" printer0
cupstestppd LEXC510.ppd
cupsaddsmb printer0
cups-config --cflags
cups-config --datadir
cups-config --ldflags
cups-config --libs
cups-config --serverbin
cups-config --serverroot
```

- **Run the CUPS Manager graphical application**
- **Enable a CUPS printer**
- **Disable a CUPS printer**
- **Accept a job sent on a printer queue**
- **Reject a job sent on a printer queue, with an informational message**
- **Test the conformance of a PPD file to the format specification**
- **Export a printer to Samba (for use with MS Windows clients)**

```
lpstat
lpadmin
lpadmin -p printer0 -P LEXC750.ppd
lp -d printer0 file
```

- **Show CUPS status information**
- **Administer CUPS printers**
- **Specify a PPD (Adobe PostScript Printer Description) file to associate to a printer**
- **Print a file on the specified printer**

```
lpq
lpq -P printer0
lpq user
lpq -P printer0 jobnumber
lpq -P printer0 user
lpq -P printer0 -
lpc
```

- **View the default print queue**
- **View a specific print queue**
- **View the print queue of a specific user**
- **Delete a specific job from a printer queue**
- **Delete all jobs from a specific user from a printer queue**
- **Delete all jobs from a printer queue**
- **Manage print queues**

```
a2ps file.txt
ps2pdf file.ps
mpage file.ps
gv file.ps
```

- **Convert a text file to PostScript**
- **Convert a file from PostScript to PDF**
- **Print a PostScript document on multiple pages per sheet on a PostScript printer**
- **View a PostScript document (the `gv` software is a derivation of GhostView)**
An IPv4 address is 32-bit long, and is represented divided in four octets (dotted-quad notation), e.g. 193.22.33.44.

There are approximately $4 \times 10^{9}$ total possible IPv4 addresses.

IPv4 classful addressing is obsolete and has been replaced by CIDR (Classless Inter-Domain Routing).

<table>
<thead>
<tr>
<th>IPv4 addressing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address range</strong></td>
</tr>
<tr>
<td>Classful</td>
</tr>
<tr>
<td>Class A (Unicast)</td>
</tr>
<tr>
<td>Class B (Unicast)</td>
</tr>
<tr>
<td>Class C (Unicast)</td>
</tr>
<tr>
<td>Class D (Multicast)</td>
</tr>
<tr>
<td>Class E (Experimental)</td>
</tr>
<tr>
<td>Private</td>
</tr>
<tr>
<td>Private Class A</td>
</tr>
<tr>
<td>Private Class B</td>
</tr>
<tr>
<td>Private Class C</td>
</tr>
<tr>
<td>Reserved</td>
</tr>
<tr>
<td>Source</td>
</tr>
<tr>
<td>Loopback</td>
</tr>
<tr>
<td>Autoconf</td>
</tr>
<tr>
<td>TEST-NET</td>
</tr>
<tr>
<td>6to4 relay anycast</td>
</tr>
<tr>
<td>Device benchmarks</td>
</tr>
</tbody>
</table>

An IPv6 address is 128-bit long, and is represented divided in eight 16-bit groups (4 hex digits). Leading zeros in each group can be deleted. A single chunk of one or more adjacent 0000 groups can be deleted. e.g. 2130:0000:0000:0000:0007:0040:15bc:235f which can also be written as 2130::7:40:15bc:235f.

There are approximately $3 \times 10^{38}$ total possible IPv6 addresses.

The IANA (Internet Assigned Numbers Authority) manages the allocation of IPv4 and IPv6 addresses, assigning large blocks to RIRs (Regional Internet Registries) which in turn allocate addresses to ISPs (Internet Service Providers) and other local registries. These address blocks can be searched via a WHOIS query to the appropriate RIR, which is:

<table>
<thead>
<tr>
<th>IANA Region</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFRINIC</td>
<td>for Africa</td>
</tr>
<tr>
<td>ARIN</td>
<td>for US, Canada, and Antarctica</td>
</tr>
<tr>
<td>APNIC</td>
<td>for Asia and Oceania</td>
</tr>
<tr>
<td>LACNIC</td>
<td>for Latin America</td>
</tr>
<tr>
<td>RIPE NCC</td>
<td>for Europe, Middle East, and Russia</td>
</tr>
</tbody>
</table>

IPv6 addressing

<table>
<thead>
<tr>
<th><strong>Address range</strong></th>
<th><strong>Prefix</strong></th>
<th><strong>Number of addresses</strong></th>
<th><strong>Reference</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td>64-bit network prefix (&gt;= 48-bit routing prefix + &lt;= 16-bit subnet id) + 64-bit interface identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A 48-bit MAC address is transformed into a 64-bit EUI-64 by inserting ff:fe in the middle.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A EUI-64 is then transformed into an IPv6 interface identifier by inverting the 7th most significant bit.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link-local</td>
<td>fe80:0000:0000:0000:0000 + 64-bit interface identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast</td>
<td>ff + 4-bit flag + 4-bit scope field + 112-bit group ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An IPv6 address is 128-bit long, and is represented divided in eight 16-bit groups (4 hex digits). Leading zeros in each group can be deleted. A single chunk of one or more adjacent 0000 groups can be deleted. e.g. 2130:0000:0000:0000:0007:0040:15bc:235f which can also be written as 2130::7:40:15bc:235f.
### VLSM chart - Last octet subnetting (CIDR notation)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>10000000</td>
<td>11000000</td>
<td>11100000</td>
<td>11110000</td>
<td>11111000</td>
<td>11111100</td>
</tr>
<tr>
<td>1 subnet</td>
<td>2 subnets</td>
<td>4 subnets</td>
<td>8 subnets</td>
<td>16 subnets</td>
<td>32 subnets</td>
<td>64 subnets</td>
</tr>
<tr>
<td>254 hosts each</td>
<td>126 hosts each</td>
<td>62 hosts each</td>
<td>30 hosts each</td>
<td>14 hosts each</td>
<td>6 hosts each</td>
<td>2 hosts each</td>
</tr>
<tr>
<td>254 total hosts</td>
<td>252 total hosts</td>
<td>248 total hosts</td>
<td>240 total hosts</td>
<td>224 total hosts</td>
<td>192 total hosts</td>
<td>128 total hosts</td>
</tr>
</tbody>
</table>

Each block of a column identifies a subnet, whose range of valid hosts addresses is \([\text{network address} + 1 \ldots \text{broadcast address} - 1]\) inclusive.

The network address of the subnet is the number shown inside a block.
The broadcast address of the subnet is the network address of the block underneath -1 or, for the bottom block, .255.
The following diagram illustrates the ISO/OSI and TCP/IP protocol stack models, along with the corresponding standards and data transmission units.

### ISO/OSI and TCP/IP protocol stack models

<table>
<thead>
<tr>
<th>Layer</th>
<th>ISO/OSI</th>
<th>TCP/IP</th>
<th>Standards</th>
<th>Data transmission unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Application</td>
<td>Application</td>
<td>HTTP, SMTP, POP, etc.</td>
<td>Message</td>
</tr>
<tr>
<td>6</td>
<td>Presentation</td>
<td>Transport</td>
<td>TCP, UDP</td>
<td>Segment (TCP), datagram (UDP)</td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
<td>Internet</td>
<td>IPV4, IPV6, ICMP, etc.</td>
<td>Packet</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td>Network</td>
<td>Ethernet, Wi-Fi, etc.</td>
<td>Frame</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
<td>Data Link</td>
<td>Bit</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Data Link</td>
<td>Physical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
<td>Network Access</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TCP connection establishment

(Two-way handshake)

1. SYN (Seq = n)
2. SYN/ACK (Seq = m, Ack = n+1)
3. ACK (Ack = m+1)

### TCP connection termination

1. FIN (Seq = p)
2. ACK (Ack = p+1)
3. FIN (Seq = q)
4. ACK (Ack = q+1)
### Wireless networking

#### Most common wireless standards

<table>
<thead>
<tr>
<th>IEEE standard</th>
<th>Known as</th>
<th>Frequency (GHz)</th>
<th>Max speed (Mbps)</th>
<th>Max range (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11</td>
<td>Wi-Fi</td>
<td>2.4</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>802.11a</td>
<td></td>
<td>5</td>
<td>54</td>
<td>100</td>
</tr>
<tr>
<td>802.11b</td>
<td></td>
<td>2.4</td>
<td>11</td>
<td>150</td>
</tr>
<tr>
<td>802.11g</td>
<td></td>
<td>2.4</td>
<td>54</td>
<td>150</td>
</tr>
<tr>
<td>802.11n</td>
<td></td>
<td>2.4, 5</td>
<td>54, 600</td>
<td>250</td>
</tr>
<tr>
<td>802.15.1</td>
<td>Bluetooth</td>
<td>2.4</td>
<td>50</td>
<td>10 - 250</td>
</tr>
<tr>
<td>802.16</td>
<td>WiMax</td>
<td>2 - 11</td>
<td>1000</td>
<td>10000</td>
</tr>
</tbody>
</table>

#### Wireless transmission techniques

<table>
<thead>
<tr>
<th>Transmission technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct-Sequence Spread Spectrum (DSSS)</td>
<td>Spread-spectrum modulation technique that modulates the original data with a pseudorandom bit sequence (spreading sequence). It is used to reduce signal interference.</td>
</tr>
<tr>
<td>Frequency-Hopping Spread Spectrum (FHSS)</td>
<td>Radio transmission technique consisting in rapidly changing the carrier frequency amongst different frequencies, in sync between transmitter and receiver. It is used to reduce signal interference, avoid eavesdropping, and allow code-division multiple access (CDMA) communications.</td>
</tr>
<tr>
<td>Orthogonal Frequency-Division Multiplexing (OFDM)</td>
<td>Digital multi-carrier modulation technique which uses multiple orthogonal subcarrier signal frequencies to transmit data, mapping information on the changes in the carrier phase, frequency, or amplitude. It is used to cope with severe channel conditions.</td>
</tr>
<tr>
<td>Multiple-Input Multiple-Output Orthogonal Frequency-Division Multiplexing (MIMO-OFDM)</td>
<td>Access mode for 4G and 5G broadband wireless communications. It is used to increase spectral efficiency and reduce signal interference.</td>
</tr>
<tr>
<td>Wireless encryption algorithms</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---</td>
</tr>
<tr>
<td><strong>WEP (Wired Equivalent Privacy)</strong></td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>WEP uses a pre-shared key with a length of 40, 104, or 232 bits, with a random 24-bit <strong>IV (Initialization Vector)</strong> added to the key. A CRC-32 checksum is computed on the data and added to it as <strong>ICV (Integrity Check Value)</strong>. WEP key and IV are fed to the RC4 stream cipher to generate a key stream, which is XORed with the data and ICV to obtain the encrypted data. WEP is insecure because of the short length of the IV, which leads to IV reuse; furthermore, the WEP standard does not even require a different IV for each packet. Key reuse in a stream cipher is bad practice and leads to weak encryption. A weak IV may even allow to deduce the WEP pre-shared key. In case of an IV collision, it is possible to reconstruct the RC4 key stream from the IV and the packet's decrypted payload. Therefore, WEP does not provide cryptographic integrity protection of a packet, and is now obsolete.</td>
<td></td>
</tr>
</tbody>
</table>

Some wireless APs use **LEAP (Lightweight Extensible Authentication Protocol)**, a Cisco proprietary version of the EAP authentication method for WLANs. LEAP can use either dynamic WEP keys (keys that change very often to minimize cracking exposure) or TKIP. LEAP uses either the MS-CHAP or the EAP-FAST authentication protocol. However, WEP with LEAP is still considered vulnerable.

| **WPA (Wi-Fi Protected Access)** | draft IEEE 802.11i |
| In WPA, the **TKIP (Temporal Key Integrity Protocol)** feeds a 128-bit temporal key and a 64-bit **MIC (Message Integrity Check)** to the RC4 stream cipher to obtain the encrypted data. It uses the CRC-32 checksum algorithm strengthened by the use of Michael MIC codes. IV size is 48 bits. TKIP adds a rekeying mechanism to provide fresh encryption and integrity keys, changing temporal keys every 10000 packets in sync between Access Point and client. |

| **WPA2 (Wi-Fi Protected Access II)** | IEEE 802.11i |
| WPA2 is encrypted using **CCMP (Counter Mode CBC-MAC Protocol)**, which utilizes AES encryption. IV size is 48 bits. **WPA2-Personal** uses a PSK (Pre-Shared Key). The Access Point encrypts the data using a 128-bit key derived from a passphrase with length from 8 to 63 characters. Encryption keys are unique for each client, and change frequently. **WPA2-Enterprise** uses centralized client authentication via 802.1X, either EAP (Extensible Authentication Protocol) or RADIUS (Remote Authentication Dial-In User Service). A TLS-encapsulated secured version of EAP, called **PEAP (Protected Extensible Authentication Protocol)**, is also available. After PSK or 802.1X authentication, a shared secret key called **PMK (Pairwise Master Key)** is generated, and is validated through a four-way handshake between wireless client and Access Point:

1. AP sends a nonce to the client, which uses it to build the **PTK (Pairwise Transient Key)**
2. The client sends a nonce and a MIC to the AP
3. The AP builds and sends the **GTK (Group Temporal Key)** with another MIC to the client
4. The client acknowledges reception to the AP |

Wireless encryption can be used in conjunction with other security measures such as SSID cloaking (security by obscurity) and MAC address filtering (whitelisting), which however are not very effective.
## Most common well-known ports

<table>
<thead>
<tr>
<th>Port number</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>TCP Daytime Protocol</td>
</tr>
<tr>
<td>20</td>
<td>TCP FTP (data)</td>
</tr>
<tr>
<td>21</td>
<td>TCP FTP (control)</td>
</tr>
<tr>
<td>22</td>
<td>TCP SSH</td>
</tr>
<tr>
<td>23</td>
<td>TCP Telnet</td>
</tr>
<tr>
<td>25</td>
<td>TCP SMTP</td>
</tr>
<tr>
<td>53</td>
<td>TCP/UDP DNS</td>
</tr>
<tr>
<td>67</td>
<td>UDP BOOTP/DHCP (server)</td>
</tr>
<tr>
<td>68</td>
<td>UDP BOOTP/DHCP (client)</td>
</tr>
<tr>
<td>69</td>
<td>TCP TFTP</td>
</tr>
<tr>
<td>80</td>
<td>TCP HTTP</td>
</tr>
<tr>
<td>88</td>
<td>TCP Kerberos</td>
</tr>
<tr>
<td>110</td>
<td>TCP POP3</td>
</tr>
<tr>
<td>119</td>
<td>TCP NNTP</td>
</tr>
<tr>
<td>123</td>
<td>UDP NTP</td>
</tr>
<tr>
<td>135</td>
<td>TCP/UDP Microsoft RPC</td>
</tr>
<tr>
<td>137</td>
<td>TCP/UDP Microsoft NetBIOS Name Service / WINS</td>
</tr>
<tr>
<td>138</td>
<td>TCP/UDP Microsoft NetBIOS Datagram Service</td>
</tr>
<tr>
<td>139</td>
<td>TCP/UDP Microsoft NetBIOS Session Service</td>
</tr>
<tr>
<td>143</td>
<td>TCP IMAP</td>
</tr>
<tr>
<td>161</td>
<td>UDP SNMP</td>
</tr>
<tr>
<td>162</td>
<td>TCP/UDP SNMP Trap</td>
</tr>
<tr>
<td>389</td>
<td>TCP/UDP LDAP</td>
</tr>
<tr>
<td>443</td>
<td>TCP HTTPS (HTTP over SSL/TLS)</td>
</tr>
<tr>
<td>445</td>
<td>TCP/UDP Microsoft SMB</td>
</tr>
<tr>
<td>465</td>
<td>TCP SMTP over SSL</td>
</tr>
<tr>
<td>500</td>
<td>UDP IPSec ISAKMP / IKE</td>
</tr>
<tr>
<td>514</td>
<td>UDP Syslog</td>
</tr>
<tr>
<td>515</td>
<td>TCP/UDP Line Printer Daemon</td>
</tr>
<tr>
<td>901</td>
<td>TCP Samba SWAT</td>
</tr>
<tr>
<td>993</td>
<td>TCP IMAPS (IMAP over SSL)</td>
</tr>
<tr>
<td>995</td>
<td>TCP POP3S (POP3 over SSL)</td>
</tr>
<tr>
<td>4500</td>
<td>UDP IPSec NAT Traversal</td>
</tr>
</tbody>
</table>

1-1023: privileged ports, used server-side
1024-65535: unprivileged ports, used client-side

The file `/etc/services` lists all well-known ports.
Network configuration - commands

- `ip a`  
  Display configuration of all network interfaces

- `ip addr`  
  Display configuration of all network interfaces

- `ip addr show`  
  Display configuration of all network interfaces

- `ifconfig -a`  
  Display configuration of all network interfaces

- `ip link show eth0`  
  Display configuration of `eth0`

- `ifconfig eth0`  
  Display configuration of `eth0`

- `ip addr add dev eth0 10.1.1.3/24`  
  Configure IP address of `eth0`

- `ifconfig eth0 10.1.1.3 netmask 255.255.255.0 broadcast 10.1.1.255`  
  Configure IP address of `eth0`

- `ifconfig eth0 hw ether 45:67:89:ab:cd:ef`  
  Configure MAC address of `eth0`

- `ip link set eth0 up`  
  Activate `eth0`

- `ifconfig eth0 up`  
  Activate `eth0`

- `ip link set eth0 down`  
  Shut down `eth0`

- `ifconfig eth0 down`  
  Shut down `eth0`

- `ifdown eth0`  
  Shut down `eth0`

- `dhclient eth0`  
  Request an IP address via DHCP

- `pump -i eth0`  
  Request an IP address via DHCP

- `dhcpcd eth0`  
  Request an IP address via DHCP

- `ip neigh`  
  Show the ARP cache table (containing mappings of MAC to IP addresses)

- `arp -a`  
  Show the ARP cache table (containing mappings of MAC to IP addresses)

- `ip neigh show 10.1.1.4`  
  Show the ARP cache entry for a host

- `arp 10.1.1.4`  
  Show the ARP cache entry for a host

- `ip neigh add 10.1.1.5 lladdr 01:23:45:67:89:ab dev eth0`  
  Add a new ARP entry for a host

- `arp -s 10.1.1.5 01:23:45:67:89:ab`  
  Add a new ARP entry for a host

- `ip neigh del 10.1.1.5 dev eth0`  
  Delete an ARP entry

- `arp -d 10.1.1.5`  
  Delete an ARP entry

- `ip neigh flush all`  
  Delete the ARP table for all interfaces

- `hostname`  
  Get the hostname

- `hostname -f`  
  Get the FQDN (Fully Qualified Domain Name)

- `hostname mybox`  
  Set the hostname

- `hostnamectl set-hostname --static "mybox"`  
  Set the hostname

- `hostnamectl`  
  Get the hostname, OS, and other information

- `/etc/init.d/networking restart`  
  Restart network services

- `/etc/init.d/network restart`  
  Restart network services

- `ethtool option device`  
  Query or control network driver and hardware settings

- `ethtool eth0`  
  View hardware settings of `eth0`

- `rdisc`  
  Network router discovery daemon. Client for IRDP (ICMP Router Discover Protocol). Runs at bootup to populate the network routing tables with default routes

- `xinetd`  
  Extended Internet services daemon. Many network services are run by it rather than standalone; `xinetd` operates as a super server, listening on all service ports defined in its configuration, and upon receiving a connection request it starts the appropriate service
Network configuration - files

/etc/hosts
Mappings between IP addresses and hostnames, for name resolution

127.0.0.1  localhost.localdomain  localhost
10.2.3.4   myhost.domain.org      myhost

/etc/nsswitch.conf
Sources that must be used by various system library lookup functions

passwd:  files nisplus nis
shadow:  files nisplus nis
group:   files nisplus nis
hosts:   files dns nisplus nis

/etc/host.conf
Sources for name resolution, for systems before glibc2.
Obsolete, superseded by /etc/nsswitch.conf

order hosts,bind
multi on

/etc/resolv.conf
Domain names that must be appended to bare hostnames, and DNS servers that will be used for name resolution

search domain1.org domain2.org
nameserver  192.168.3.3
nameserver  192.168.4.4

/etc/networks
Mappings between network addresses and names

loopback  127.0.0.0
mylan     10.2.3.0

/etc/services
List of service TCP/UDP port numbers

/etc/protocols
List of available protocols

/sys/class/net
List of all network interfaces in the system
## Network configuration - distro-specific files

### Red Hat

- **/etc/sysconfig/network**
  - Network configuration file
  - ADDRESS=10.2.3.4
  - NETMASK=255.255.255.0
  - GATEWAY=10.2.3.254
  - HOSTNAME=mylinuxbox.example.org
  - NETWORKING=yes

- **/etc/sysconfig/network-scripts/ifcfg-eth0**
  - Configuration file for eth0.
  - This file is read by the `ifup` and `ifdown` scripts
  - DEVICE=eth0
  - TYPE=Ethernet
  - BOOTPROTO=none
  - ONBOOT=yes
  - NM_CONTROLLED=no
  - IPADDR=10.2.3.4
  - NETMASK=255.255.255.0
  - GATEWAY=10.2.3.254
  - DNS1=8.8.8.8
  - DNS2=4.4.4.4
  - USERCTL=no

- **/etc/sysconfig/network-scripts/ifcfg-eth0:0 /etc/sysconfig/network-scripts/ifcfg-eth0:1 /etc/sysconfig/network-scripts/ifcfg-eth0:2**
  - Multiple configuration files for a single eth0 interface, which allows binding multiple IP addresses to a single NIC

- **/etc/sysconfig/network-scripts/route-eth0**
  - Static route configuration for eth0
  - default 10.2.3.4 dev eth0
  - 10.7.8.0/24 via 10.2.3.254 dev eth0
  - 10.7.9.0/24 via 10.2.3.254 dev eth0

- **/etc/ethertypes**
  - Ethernet frame types.
  - Lists various Ethernet protocol types used on Ethernet networks

### Debian

- **/etc/network/interfaces**
  - List and configuration of all network interfaces
  - allow-hotplug eth0
  - iface eth0 inet static
  - address 10.2.3.4
  - netmask 255.255.255.0
  - gateway 10.2.3.254
  - dns-domain example.com
  - dns-nameservers 8.8.8.8 4.4.4.4

- **/etc/hostname**
  - Hostname of the local machine

- **/etc/ethers**
  - ARP mappings
In RHEL7 and later the network configuration is managed by the NetworkManager daemon. A **connection** is a network configuration that applies to a **device** (aka network interface). A device can be included in multiple connections, but only one of them may be active at a time. The configuration for **connection** is stored in the file `/etc/sysconfig/network-scripts/ifcfg-connection`. Although it is possible to set up networking by editing these configuration files, it is much easier to use the command `nmcli`.

```
nmcli device status  # Show all network devices
nmcli device disconnect iface  # Disconnects the device iface. This command should be used instead of nmcli connection down connection because if connection is set to autoconnect, Network Manager will bring the connection (and the device) up again short time later

nmcli connection show  # Show all connections. Connections with an empty device entry are inactive
nmcli connection show --active  # Show active connections
nmcli connection show connection  # Show the configuration of connection
nmcli connection add con-name connection \  # Configure a new connection that uses the Ethernet interface iface and assigns it an IPv4 address and gateway
type ethernet ifname iface ipv4.method manual \  ipv4.addresses 10.0.0.13/24 ipv4.gateway 10.0.0.254
nmcli connection modify connection options  # Modify the configuration of connection
nmcli connection up connection  # Brings up a connection
nmcli connection reload  # Reload any manual change made to the files /etc/sysconfig/network-scripts/ifcfg-*
```

The manpage `man nmcli-examples` contains examples of network configuration.
Network teaming allows binding together two or more network interfaces to increase throughput or provide redundancy. RHET and later implement network teaming via the `teamd` daemon.

**How to set up a teaming connection**

1. `nmcli connection add type team con-name teamcon ifname teamif \ config ""runner":{"name":"loadbalance"}"`

2. `nmcli connection modify teamcon ipv4.method manual \ ipv4.addresses 10.0.0.14/24 ipv4.gateway 10.0.0.254`

3. `nmcli connection add type team-slave ifname iface \ master teamcon`

4. Repeat the previous step for each slave interface.

```
setdctl teamif state
```
Show the state of the team interface `teamif`

```
teamnl teamif command
```
Debug a team interface `teamif`

A network bridge emulates a hardware bridge, i.e. a Layer 2 device able to forward traffic between networks based on MAC addresses.

**How to set up a bridge connection**

1. `nmcli connection add type bridge con-name brcon ifname brif`

2. `nmcli connection modify brcon ipv4.method manual \ ipv4.addresses 10.0.0.15/24 ipv4.gateway 10.0.0.254`

3. `nmcli connection add type bridge-slave ifname iface \ master brcon`

4. Repeat the previous step for each slave interface.

```
brctl show brif
```
Display information about the bridge interface `brif`

The manpage `man teamd.conf` contains examples of team configurations and runners. The manpage `man nmcli-examples` contains examples of teaming and bridging configuration.
Network tools

dig example.org

Performs a DNS lookup for the specified domain or hostname. Returns information in BIND zone file syntax; uses an internal resolver and hence does not honor /etc/resolv.conf.

host example.org

Performs a DNS lookup for the specified domain or hostname. Does honor /etc/resolv.conf.

nslookup example.org (deprecated)

Performs a DNS lookup for the specified domain or hostname. Does honor /etc/resolv.conf.

dig @nameserver.org -t MX example.org

Performs a DNS lookup for the MX record of the specified domain, querying nameserver.

dig example.org any

Gets all DNS records for a domain.

dig -x a.b.c.d

dig @nameserver.org -t MX example.org

these commands can be used to find what is the maximum frame size allowed on the network, by trying increasingly higher values for size until it exceeds the MTU and the datagram is unable to reach the destination host.

whois example.org

Queries the WHOIS service for an Internet resource (usually a domain name).

ping host

Tests if a remote host can be reached and measure the round-trip time to it. This is done by sending an ICMP Echo Request datagram and awaiting an ICMP Echo Response.

ping -M do -s size host

Pings a remote host using an ICMP packet of size size (default is 56 bytes) and setting the DF (Don’t Fragment) bit. This command can be used to find what is the maximum frame size allowed on the network, by trying increasingly higher values for size until it exceeds the MTU and the datagram is unable to reach the destination host.

fping -a host1 host2 host3

Pings multiple hosts in parallel and report which ones are alive.

bing host1 host2

Calculates point-to-point throughput between two hosts.

traceroute host

Prints the route, hop by hop, packets trace to a remote host. This is done by sending a sequence of ICMP Echo Request datagrams with increasing TTL values, starting with TTL=1, and expecting ICMP Time Exceeded datagrams.

tracepath host

Simpler traceroute.

mtr host

traceroute and ping combined.

telnet host

telnet host port

Establishes a telnet connection to a remote host.

uucp srchost!path desthost!path

Unix-to-Unix copies. Copies files between hosts, identified by a bang path. Obsolete

wget

Downloads a file via HTTP, HTTPS, or FTP.

wget --no-clobber --html-extension --page-requisites --convert-links --recursive --domains example.org --no-parent www.example.org/path

Download a whole website www.example.org/path.
### Advanced network tools

**redir --laddr=ip1 --lport=port1 \ 
--caddr=ip2 --cport=port2**

Redirect all connections, coming to local IP address `ip1` and port `port1`, to remote IP address `ip2` and port `port2`.

**stunnel**

TLS encryption wrapper. Can be used to secure any client-server protocol.

**socat**

Establish two bidirectional data stream and transfer data between them.

**socat TCP-LISTEN:80,fork TCP:host:80**

Forward local HTTP port to remote host's HTTP port.

**socat TCP:timeserver:13 -**

Query a `timeserver` using the Daytime Protocol.

**curl**

Transfer data to or from a remote host via HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, DICT, TELNET, LDAP, or FILE.

**curl www.example.org/file -o myfile**

Download a file via HTTP and save it locally under another name.

**curl -u user:psw 'ftp://server/file'**

Download a file via FTP, after logging in to the server.

**curl -XPUT webserver -d'data'**

Send an HTTP PUT command with `data` to `webserver`.

**inetsim**

Simulate Internet services. This is useful when setting up a confined lab for malware analysis.

**pktgen**

Network packet generator. Uses the DPDK (Data Plane Development Kit) packet processing framework.

**trafgen**

Network packet generator.

**packETH**

Ethernet packet generator (GUI).

**packETHcli**

Ethernet packet generator (command line).
Wireless network tools

iwlist wlan0 scan
List all wireless devices in range, with their quality of signal and other information

iwlist wlan0 freq
Display transmission frequency settings

iwlist wlan0 rate
Display transmission speed settings

iwlist wlan0 txpower
Display transmission power settings

iwlist wlan0 key
Display encryption settings

iwgetid wlan0 option
Print NWID, ESSID, AP/Cell address or other information about the wireless network that is currently in use

iwconfig wlan0
Display configuration of wireless interface wlan0

iwconfig wlan0 option
Configure wireless interface wlan0

iw dev wlan0 station dump
On a wireless card configured in AP Mode, display information (e.g. MAC address, tx/rx, bitrate, signal strength) about the clients

rfkill list
List installed wireless devices

rfkill unblock n
Enable wireless device number n

hostapd
Daemon that allows a wireless card to function in Host AP Mode, i.e. perform all functions of an Access Point

hcidump -i device
Display raw HCI (Host Controller Interface) data exchanged with a Bluetooth device
Network monitoring tools

**netstat**
Display current network connections.
Options:
- `-t` Display active TCP connections
- `-l` Display only listening sockets
- `-a` Display all listening and non-listening sockets
- `-p` Display PID and name of program to which each socket belongs
- `-i` Display network interfaces
- `-s` Display protocol statistics
- `-r` Display kernel routing tables (equivalent to `route -e`)
- `-n` Do not resolve hostnames or portnames
- `-c` Continuously display connections

**ss**
Display socket statistics (similarly to `netstat`)
**ss -t -a**
Display all TCP sockets

**arp-scan**
Scan all hosts on the current LAN. Uses ARP (Layer 2) packets; therefore it is able to also find hosts configured to drop all IP or ICMP traffic, and it cannot scan hosts outside the LAN

**snoop** *(Solaris)*
Packet sniffer

**tcptrace**
Tool for the analysis of TCP dump files such as those generated by `tcpdump`, `snoop`, etc.

**ipgrab**
Packet sniffer that includes full header fields

**dhcpdump**
DHCP packet sniffer

**ngrep**
Filter data payload of network packets matching a specified regex

**nload**
Display a graph of the current network usage

**ntop**
Network usage analyzer

**ntopng**
Network usage analyzer

**nethogs**
Display bandwidth network usage by process

**iptraf**
Interactive IP LAN monitor (ncurses UI)

**iptraf-ng**
Interactive IP LAN monitor (ncurses UI)

**netserver**
Run a network performance benchmark server

**netperf**
Do network performance benchmarks by connecting to a netserver server

**iperf -s**
Run a network throughput benchmark server

**iperf -c server**
Perform network throughput tests in client mode, by connecting to an iperf server
Nmap is a network analyzer, auditing tool, and penetration testing tool. The GUI equivalent is Zenmap.

**nmap options host**

Scan a host, or all hosts in a subnet

<table>
<thead>
<tr>
<th>Port state</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td>An application is listening for connections on the port</td>
</tr>
<tr>
<td>closed</td>
<td>No application is listening for connections on the port</td>
</tr>
<tr>
<td>filtered</td>
<td>Port is not responding to probe due to a firewall blocking the port, so port may be open or closed</td>
</tr>
<tr>
<td>unfiltered</td>
<td>Port is responding to probe, but it is impossible to tell whether port is open or closed</td>
</tr>
</tbody>
</table>

**Scan technique options**

- **-sT** TCP connect scan aka full-open scan. Completes the three-way handshake; response will be SYN/ACK if port is open, RST if port is closed. Slow and likely to trigger IDS
- **-sS** SYN scan aka stealth scan or half-open scan. Sends a TCP packet with SYN flag set; same response as the TCP connect scan. Fast and stealth
- **-sN** Null scan. Sends a TCP packet with no flag set; response will be none if port is open, RST if port is closed. Stealth. Works only if the target host’s OS TCP/IP implementation is based on RFC 793
- **-sF** FIN scan. Sends a TCP packet with FIN flag set. Same response as null scan. Scanning using a TCP packet with FIN and/or PSH and/or URG flags set is known as inverse TCP flag scan.
- **-sX** Xmas tree scan. Sends a TCP packet with FIN and/or PSH and/or URG flags set. Same response as null scan
- **-sA** ACK scan. Sends a TCP packet with ACK flag set; response will be RST if port is open or closed (unfiltered), no response or ICMP error if port is filtered. Further analysis can be done on the TCP/IP RST response packet: if the TTL field is less than the max value, or the Window Size field is nonzero, port is open. Used to discover firewall rules, and to determine firewall type: if unfiltered ports are reported the firewall is stateless, otherwise the firewall is stateful
- **-sI** zombiehost IP ID idle scan. Uses a zombie host to perform the scan and predicts the port state of the target host by analyzing the IP fragmentation ID sequence numbers from the zombie host; if the sequence number has increased by 2 port is open, if it has increased by 1 port is closed. Fully stealth as no packets are sent from the scanning machine to the target host
- **-sU** UDP scan. Sends a UDP packet; response will be none if port is open or filtered, ICMP Port Unreachable (Type 3 Destination Unreachable) error if port is closed
- **-sO** IP protocol scan. Cycles through IP protocol numbers (instead of TCP or UDP ports) to try to determine which IP protocols (TCP, ICMP, IGMP, etc.) the target host supports; response will be any protocol response or none if port is open, ICMP Protocol Unreachable (Type 3 Destination Unreachable) error if port is closed, other ICMP Type 3 Destination Unreachable errors or none if port is filtered
- **-sR** RPC scan. Floods all TCP/UDP ports found open with SunRPC program NULL commands to try to determine whether they are RPC ports, and if yes, the service program and version number. It is recommended to use -sV instead as it gives more information
- **--script=name** Script scan. Uses the default script set. The Nmap Scripting Engine permits writing scripts (in the Lua programming language) to perform automatically various types of network scans
  - **--script=sniffer-detect host** Check if a host has its NIC in promiscuous mode (sniffer)
  - **--script=firewalk host** Attempt to detect firewall or gateway rules
  - **--script=http-trace -d host** Send an HTTP TRACE request to find if TRACE method is enabled
  - **--script=http-enum host** Enumerate dirs used by common web applications and webservers
### Nmap Options

#### Host Discovery Options

- `-sL` **List scan.** Simple network host discovery, with reverse DNS resolution. No packet is sent to target hosts.
- `-sn` **No port scan aka ping sweep.** Used to detect how many hosts are up.
- `-Pn` No ping; skip host discovery. The subsequent scan operation will be performed against all hosts, instead of only those discovered to be up during this phase.
- `-PS` Send a TCP SYN packet to the specified port.
- `-PA` Send a TCP ACK packet to the specified port.
- `-PU` Send a UDP packet to the specified port.
- `-PY` Send a SCTP packet containing a minimal INIT chunk to the specified port.
- `-PE` Send a ICMP Echo Request (type 8).
- `-PP` Send a ICMP Timestamp Request (type 13).
- `-PM` Send a ICMP Address Mask Request (type 17).
- `-PO` Send IP packets with the specified protocol number set in their header.
- `-PR` ARP scan. Default discovery type when scanning the current LAN.
- `--traceroute` After the scan, trace path to host to determine port and protocol most likely to reach the target host.

#### IDS Evasion, Firewall Evasion, and Spoofing Options

- `-f` Use tiny fragmented packets (8 bytes or less) for IDS evasion. Might crash the target host.
- `--mtu offset` Use fragmented packets of size `offset` (must be a multiple of 8) for IDS evasion.
- `-D ip` Spoof the scanning machine IP address as `ip`.
- `-D RND:n` Spoof the scanning machine IP address using `n` randomly generated addresses. The real IP address is included among the decoys.
- `--ip-options “L ip1 ip2”` Use loose source routing for IDS evasion, requiring that the packet is loose source routed through the waypoints with IP address `ip1` and `ip2`.
- `--ip-options “S ip1 ip2”` Use strict source routing for IDS evasion, requiring that the packet is strictly source routed through the waypoints with IP address `ip1` and `ip2`. All waypoints must be specified.

#### Timing Options

- `-T0` Paranoid. Extremely slow serialized scan for IDS evasion. Will take a long time to complete.
- `-T1` Sneaky. Very slow serialized scan for IDS evasion. Will take a long time to complete.
- `-T2` Polite. Slow serialized scan to consume less bandwidth and resources of the target.
- `-T3` Normal. Parallel scan. Default.
- `-T4` Aggressive. Fast parallel scan, to be used on networks with a high bandwidth. Recommended.
- `-T5` Insane. Very fast parallel scan, to be used on networks with a very high bandwidth. Might be less accurate.

#### Other Options

- `-A` Aggressive scan. Equivalent to `-O -sV -sC --traceroute`.
- `-O` OS fingerprinting, to find out which operating system is running on target host.
- `-sV` Version detection, to determine protocol, application name, version number, device type, etc.
- `-6` Enable IPv6 scanning.
- `-p port` Scan only the specified port or port range, instead of the most common 1000 ports for each protocol.
- `--top-ports n` Scan only the `n` most popular ports.
- `-F` Fast mode; scan fewer ports than the default, hence enumerating all hosts faster.
- `-r` Scan ports in numerical order, instead of random order.
- `-n` Do not do DNS resolution.
- `-R` Always do DNS resolution.
- `-oN file.nmap` Save output to `file` in standard format (slightly different from interactive mode output).
- `-oX file.xml` Save output to `file` in XML format.
Tcpdump is a packet sniffer (aka packet analyzer) which uses the libpcap library for packet capture. The GUI equivalent of tcpdump is Wireshark, originally called Ethereal.

Sniffers operate at the Data Link layer (Layer 2). On a wired medium, for a sniffing machine to be able to capture all network traffic, and not only the traffic from/to the machine itself, the machine’s NIC must be set to promiscuous mode. Furthermore, only traffic within a network segment connected via a hub (i.e. the collision domain) can be sniffed; in the case of a switched network, the sniffing machine needs to be connected to the switch’s SPAN port (which performs port mirroring) in order to be able to capture all traffic.

In the case of a wireless NIC, the chipset also determines capabilities for modes of operation. Active sniffing refers to sniffing through a switch. Passive sniffing refers to sniffing through a hub.

`tcpdump options expression`  
Print the content of sniffed packets that match `expression`.  
Options:  
- `-v` `-vv` Increasing levels of verbosity  
- `-n` Do not perform DNS resolution on host addresses  
- `-nn` Do not convert protocol and port numbers to names

`tcpdump -i eth0`  
Sniff all network traffic on interface `eth0`

`tcpdump ip host 10.0.0.2 tcp port 25`  
Sniff network packets on TCP port 25 from and to 10.0.0.2

`tcpdump ether host ’45:67:89:ab:cd:ef’`  
Sniff traffic from and to the network interface having MAC address 45:67:89:ab:cd:ef

`tcpdump ’src host 10.0.0.2 and (tcp port 80 or tcp port 443)’`  
Sniff HTTP and HTTPS traffic having as source host 10.0.0.2

`tcpdump -i eth0 not port 22`  
Sniff all traffic on `eth0` except that belonging to a SSH connection

`tcpdump -i eth0 arp`  
Sniff ARP traffic on `eth0`

`tcpdump ip host 10.0.0.2 and not 10.0.0.9`  
Sniff IP traffic between 10.0.0.2 and any other host except 10.0.0.9

---

### PCAP filter syntax

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tcp.port==25 or icmp</code></td>
<td>Show SMTP and ICMP traffic</td>
</tr>
<tr>
<td><code>ip.addr==10.0.0.2</code></td>
<td>Show traffic from and to 10.0.0.2</td>
</tr>
<tr>
<td><code>ip.src==10.0.0.2 or ip.dst==10.0.0.2</code></td>
<td>Show packets coming from 10.0.0.2 with frame length higher than 400</td>
</tr>
<tr>
<td><code>ip.src==10.0.0.3 and frame.pkt_len &gt; 400</code></td>
<td>Show HTTP requests</td>
</tr>
<tr>
<td><code>http.request</code></td>
<td>Show UDP packets containing the 2-byte hex sequence 0x76, 0x54 in the header or the payload, at any offset</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>==</code></td>
<td>Equal to</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Greater than</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>Not equal to</td>
</tr>
<tr>
<td><code>&lt;</code></td>
<td>Less than</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Less than or equal to</td>
</tr>
<tr>
<td><code>&amp;&amp;</code></td>
<td>Logical AND</td>
</tr>
<tr>
<td>`</td>
<td></td>
</tr>
</tbody>
</table>
Netcat is "the Swiss Army knife of networking", a very flexible generic TCP/IP client/server. Depending on the distribution, the binary is called nc, ncat (Red Hat), or netcat (SUSE).

```
nc -z 10.0.0.7 22
ncat 10.0.0.7 22

nc -l -p 25

nc 10.0.0.7 389 < file

echo "GET / HTTP/1.0\r\n\r\n" | nc 10.0.0.7 80

while true; do
  do nc -l -p 80 -q 1 < page.html; done

while true; do
echo "<html><body>Hello</body></html>" | ncat -l -p 80; done

nc -v -n -z -w1 -r 10.0.0.7 1-1023

echo "" | nc -v -n -w1 10.0.0.7 1-1023
```

- Scan for a listening SSH daemon on remote host 10.0.0.7
- Listen for connections on port 25 (i.e. mimic a SMTP server). Send any input received on stdin to the connected client and dump on stdout any data received from the client
- Push the content of file to port 389 on remote host 10.0.0.7
- Connect to web server 10.0.0.7 and issue a HTTP GET
- Start a minimal web server, serving the specified HTML page to clients
- Run a TCP port scan against remote host 10.0.0.7. Probes randomly all privileged ports with a 1-second timeout, without resolving service names, and with verbose output
- Retrieve the greeting banner of any network service that might be running on remote host 10.0.0.7
hp3 is a packet crafting tool, able to send any custom TCP/IP packet to a remote host and display the reply. It is an extension of hping2, and is command-line compatible with it while having extended capabilities for packet generation.

**hp3 options host**

Send a crafted packet to host. By default, it sends TCP headers to port 0 of remote host with no TCP flag set and a window size of 64.

<table>
<thead>
<tr>
<th>hp3 options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c n</td>
<td>Send n packets</td>
</tr>
<tr>
<td>-p n</td>
<td>Use port n</td>
</tr>
<tr>
<td>-a src</td>
<td>Set src as a fake IP source address for sent packets</td>
</tr>
<tr>
<td>--spoof src</td>
<td></td>
</tr>
<tr>
<td>-l --ICMP</td>
<td>Use the ICMP protocol. By default, hping3 uses TCP</td>
</tr>
<tr>
<td>-2 --UDP</td>
<td>Use the UDP protocol</td>
</tr>
<tr>
<td>-S n1-n2</td>
<td>Operate in scan mode, scanning the port range from n1 to n2</td>
</tr>
<tr>
<td>--scan n1-n2</td>
<td></td>
</tr>
<tr>
<td>-9 signature</td>
<td>Operate in listening mode, trying to intercept signature</td>
</tr>
<tr>
<td>--listen signature</td>
<td></td>
</tr>
<tr>
<td>-A</td>
<td>Set the ACK flag in probe packets. An ACK scan can be used to check if the remote host is alive, when it does not respond to ping packets</td>
</tr>
<tr>
<td>-S</td>
<td>Set the SYN flag in probe packets</td>
</tr>
<tr>
<td>-F</td>
<td>Set the FIN flag in probe packets</td>
</tr>
<tr>
<td>-P</td>
<td>Set the PSH flag in probe packets</td>
</tr>
<tr>
<td>-U</td>
<td>Set the URG flag in probe packets</td>
</tr>
<tr>
<td>-Q</td>
<td>Collect all TCP sequence numbers generated by the remote host</td>
</tr>
<tr>
<td>--tcp-timestamp</td>
<td>Attempt to guess the timestamp update frequency and uptime of the remote host</td>
</tr>
</tbody>
</table>

**hp3 examples**

- `hp3 -S -p 25 -c 5 host`
  Send 5 TCP packets, with the SYN flag set, to port 25 of remote host

- `hp3 --scan 1-1024 -S host`
  Perform a SYN scan on ports 1 to 1024 against the remote host

- `hp3 --udp --rand-source --data 512 host`
  Send UDP packets with random source address and a data body size of 512 bytes

- `hp3 -S -p 80 --flood host`
  Perform a TCP SYN flood DoS attack against a webserver

- `hp3 -A -p 25 host`
  Verify if a mailserver is alive (if it is, it will reply with an RST)
TCP Wrapper

The TCP Wrapper feature provides basic traffic filtering of incoming network connections. To use this feature, the service binary must have been compiled with the `libwrap.a` library.

```
ldd service_binary | grep libwrap
```

Find if a network service is TCP Wrapped

```
/etc/hosts.allow
/etc/hosts.deny
```

Host access control files used by the TCP Wrapper system.

Each file contains zero or more `daemon:client` lines. The first matching line is considered.

Access is granted when a `daemon:client` pair matches an entry in `/etc/hosts.allow`. Otherwise, access is denied when a `daemon:client` pair matches an entry in `/etc/hosts.deny`. Otherwise, access is granted.

<table>
<thead>
<tr>
<th>/etc/hosts.allow and /etc/hosts.deny lines syntax</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL: ALL</td>
<td>All services to all hosts</td>
</tr>
<tr>
<td>ALL: .example.edu</td>
<td>All services to all hosts of the example.edu domain</td>
</tr>
<tr>
<td>ALL: .example.edu EXCEPT host1.example.edu</td>
<td>All services to all hosts of example.edu, except host1</td>
</tr>
<tr>
<td>in.fingerd: .example.com</td>
<td>Finger service to all hosts of example.com</td>
</tr>
<tr>
<td>in.tftpd: LOCAL</td>
<td>TFTP to hosts of the local domain only</td>
</tr>
<tr>
<td>sshd: 10.0.0.3 10.0.0.4 10.1.1.0/24</td>
<td>SSH to the hosts and network specified</td>
</tr>
<tr>
<td>sshd: 10.0.1.0/24</td>
<td>SSH to 10.0.1.0/24</td>
</tr>
<tr>
<td>sshd: 10.0.1.0/255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>in.tftpd: ALL: spawn (/safe_dir/safe_finger -l @%h \</td>
<td>Send a finger probe to hosts attempting TFTP and notify the root user via email</td>
</tr>
<tr>
<td></td>
<td>/bin/mail -s %d-%h root) &amp;</td>
</tr>
<tr>
<td>portmap: ALL: {echo Illegal RPC request from %h \</td>
<td>When a client attempts an RPC request via portmapper (NFS access), echo a message to the terminal and notify the root user via email</td>
</tr>
<tr>
<td></td>
<td>/bin/mail root) &amp;</td>
</tr>
</tbody>
</table>
### Output of command `route -en`

<table>
<thead>
<tr>
<th>Destination</th>
<th>network or host</th>
<th>destination network or host</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>host</td>
<td>gateway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>no gateway needed, network is directly connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rejected route</td>
</tr>
<tr>
<td>Genmask</td>
<td>network mask</td>
<td>network mask to apply for the destination network</td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>destination host</td>
<td>default route</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td></td>
<td>default route</td>
</tr>
<tr>
<td>Flags</td>
<td>U</td>
<td>route is up</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>use gateway</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>target is host</td>
</tr>
<tr>
<td></td>
<td>!</td>
<td>rejected route</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>dynamically installed by daemon</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>modified from routing daemon</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>reinstate route for dynamic routing</td>
</tr>
</tbody>
</table>

|                |                |                |                |                |
|----------------|----------------|----------------|----------------|
| ip route       | Display IP routing table |
| route -en      |                |                |                |
| route -F       |                |                |                |
| netstat -rn    |                |                |                |
|                |                |                |                |
| ip route show cache | Display kernel routing cache |
| route -C       |                |                |                |
|                |                |                |                |
| ip route add default via 10.1.1.254 | Add a default gateway 10.1.1.254 |
| route add default gw 10.1.1.254 |                |                |                |
|                |                |                |                |
| ip route add 10.2.0.1 dev eth0 | Add a route for a host 10.2.0.1 |
| ip route add 10.2.0.1 via 10.2.0.254 |                |                |                |
| route add -host 10.2.0.1 gw 10.2.0.254 |                |                |                |
|                |                |                |                |
| ip route add 10.2.0.0/16 via 10.2.0.254 | Add a route for a network 10.2.0.0/16 |
| route add -net 10.2.0.0 netmask 255.255.0.0 gw 10.2.0.254 |                |                |                |
|                |                |                |                |
| ip route delete 10.2.0.1 dev eth0 | Delete a route for a host 10.2.0.1 |
| route del -host 10.2.0.1 gw 10.2.0.254 |                |                |                |
|                |                |                |                |
| ip route flush all | Delete the routing table for all interfaces |
|
The Netfilter framework provides firewalling capabilities in Linux. It is implemented by the user-space application programs \texttt{iptables} for IPv4 (which replaced \texttt{ipchains}, which itself replaced \texttt{ipfwadm}) and \texttt{ip6tables} for IPv6. \texttt{iptables} is implemented in the kernel and therefore does not have a daemon process or a service. The ability to track connection state is provided by the \texttt{ip_conntrack} kernel module.

In RHEL 6, the service \texttt{iptables} provides all firewall functionalities; the GUI frontend is \texttt{system-config-firewall}. In RHEL 7, the firewall is managed by the \texttt{firewalld} daemon, which uses \texttt{iptables} as backend. It is possible, but discouraged, to use \texttt{iptables} directly by disabling \texttt{firewalld} and installing the package \texttt{iptables-services}, which provides systemd units for \texttt{iptables}.

In RHEL 8, the firewall is managed by \texttt{firewalld}, with \texttt{nftables} (a replacement for \texttt{iptables}) as backend.

In Ubuntu, firewall capabilities are provided by the \texttt{ufw} (Uncomplicated Firewall) service, with \texttt{iptables} as backend.

\begin{verbatim}
/etc/sysconfig/iptables

Default file containing the firewall rules

iptables-restore < file
Load into iptables the firewall rules specified in the file

iptables-save > file
Save into iptables the firewall rules specified in the file

iptables rules file

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT

Delete all rules and open the firewall to all connections
\end{verbatim}
Iptables uses **tables** containing sets of **chains**, which contain sets of **rules**. Each rule has a **target** (e.g. ACCEPT). The "filter" table contains chains INPUT, FORWARD, OUTPUT (built-in chains); this is the default table to which all iptables commands are applied, unless another table is specified via the `-t` option. The "nat" table contains chains PREROUTING, OUTPUT, POSTROUTING. The "mangle" table contains chains PREROUTING, OUTPUT.

When a packet enters the system, it is handed to the INPUT chain. If the destination is local, it is processed; if the destination is not local and IP forwarding is enabled, the packet is handed to the FORWARD chain, otherwise it is dropped. An outgoing packet generated by the system will go through the POSTROUTING chain.

If NAT is in use, an incoming packet will pass at first through the PREROUTING chain, and an outgoing packet will pass last through the POSTROUTING chain.

```bash
iptables -A INPUT -s 10.0.0.6 -j ACCEPT
iptables -A INPUT -s 10.0.0.7 -j REJECT
iptables -A INPUT -s 10.0.0.8 -j DROP
iptables -A INPUT -s 10.0.0.9 -j LOG

iptables -D INPUT -s 10.0.0.9 -j LOG
iptables -D INPUT 42
iptables -F INPUT
iptables -t mangle -F
iptables -t mangle -x

iptables -L INPUT
iptables -L -n

iptables -N mychain
iptables -P INPUT DROP

iptables -A OUTPUT -d 10.7.7.0/24 -j DROP
iptables -A FORWARD -i eth0 -o eth1 -j LOG
iptables -A INPUT -p tcp --dport 1024:65535 -j ACCEPT
iptables -A INPUT --port tcp --dport 53
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -m state --state NEW -j ACCEPT
iptables -A INPUT -m state --state RELATED -j ACCEPT
iptables -A INPUT -m state --state INVALID -j ACCEPT
```

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iptables -A INPUT -s 10.0.0.6 -j ACCEPT</code></td>
<td>Add a rule to accept all packets from 10.0.0.6</td>
</tr>
<tr>
<td><code>iptables -A INPUT -s 10.0.0.7 -j REJECT</code></td>
<td>Add a rule to reject all packets from 10.0.0.7 and send back an ICMP response to the sender</td>
</tr>
<tr>
<td><code>iptables -A INPUT -s 10.0.0.8 -j DROP</code></td>
<td>Add a rule to silently drop all packets from 10.0.0.8</td>
</tr>
<tr>
<td><code>iptables -A INPUT -s 10.0.0.9 -j LOG</code></td>
<td>Add a rule to log (via syslog) all packets from 10.0.0.9</td>
</tr>
<tr>
<td><code>iptables -D INPUT -s 10.0.0.9 -j LOG</code></td>
<td>Delete a specific rule</td>
</tr>
<tr>
<td><code>iptables -D INPUT 42</code></td>
<td>Delete rule 42 of the INPUT chain</td>
</tr>
<tr>
<td><code>iptables -F INPUT</code></td>
<td>Flush all rules of the INPUT chain</td>
</tr>
<tr>
<td><code>iptables -t mangle -F</code></td>
<td>Flush all rules of the &quot;mangle&quot; table</td>
</tr>
<tr>
<td><code>iptables -t mangle -x</code></td>
<td>Delete all user-defined (not built-in) rules in the &quot;mangle&quot; table</td>
</tr>
<tr>
<td><code>iptables -L INPUT</code></td>
<td>List the rules of the INPUT chain</td>
</tr>
<tr>
<td><code>iptables -L -n</code></td>
<td>List all rules, without translating numeric values (IP addresses to FQDNs and port numbers to services)</td>
</tr>
<tr>
<td><code>iptables -N mychain</code></td>
<td>Define a new chain</td>
</tr>
<tr>
<td><code>iptables -P INPUT DROP</code></td>
<td>Define the chain policy target, which takes effect when no rule matches and the end of the rules list is reached</td>
</tr>
<tr>
<td><code>iptables -A OUTPUT -d 10.7.7.0/24 -j DROP</code></td>
<td>Add a rule to drop all packets with destination 10.7.7.0/24</td>
</tr>
<tr>
<td><code>iptables -A FORWARD -i eth0 -o eth1 -j LOG</code></td>
<td>Add a rule to log all packets entering the system via eth0 and exiting via eth1</td>
</tr>
<tr>
<td><code>iptables -A INPUT -p tcp --dport 1024:65535 -j ACCEPT</code></td>
<td>Add a rule to drop all incoming UDP traffic (protocol numbers are defined in /etc/protocols)</td>
</tr>
<tr>
<td><code>iptables -A INPUT --port tcp --dport 53</code></td>
<td>Add a rule to accept all packets coming from any unprivileged port and with destination port 53</td>
</tr>
<tr>
<td><code>iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT</code></td>
<td>Add a rule to accept incoming pings through eth0 at a maximum rate of 1 ping/second</td>
</tr>
<tr>
<td><code>iptables -A INPUT -m state --state NEW -j ACCEPT</code></td>
<td>Load the module for stateful packet filtering, and add a rule to accept all packets that are part of a communication already tracked by the state module</td>
</tr>
<tr>
<td><code>iptables -A INPUT -m state --state RELATED -j ACCEPT</code></td>
<td>Add a rule to accept all packets that are not part of a communication already tracked by the state module</td>
</tr>
<tr>
<td><code>iptables -A INPUT -m state --state INVALID -j ACCEPT</code></td>
<td>Add a rule to accept all packets that are related (e.g. ICMP responses to TCP or UDP traffic) to a communication already tracked by the state module</td>
</tr>
<tr>
<td><code>iptables -A INPUT -m state --state INVALID -j ACCEPT</code></td>
<td>Add a rule to accept all packets that do not match any of the states above</td>
</tr>
</tbody>
</table>
SNAT (Source Network Address Translation)

iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth1 \ -j SNAT --to-source 93.184.216.119

iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth1 \ -j SNAT --to-source 93.184.216.119:93.184.216.127

iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

Map all traffic leaving the LAN to the external IP address 93.184.216.119

Map all traffic leaving the LAN to a pool of external IP addresses 93.184.216.119-127

Map all traffic leaving the LAN to the address dynamically assigned to eth1 via DHCP

DNAT (Destination Network Address Translation)

iptables -t nat -A PREROUTING -i eth1 -d 93.184.216.119 \ -j DNAT --to-destination 10.0.0.13

Allow the internal host 10.0.0.13 to be publicly reachable via the external address 93.184.216.119

PAT (Port Address Translation)

iptables -t nat -A PREROUTING -i eth1 -d 93.184.216.119 \ -p tcp --dport 80 -j DNAT --to-destination 10.0.0.13:8080

iptables -t nat -A PREROUTING -i eth0 ! 10.0.0.0/24 \ -p tcp --dport 80 -j REDIRECT --to-ports 3128

Make publicly accessible a webserver that is located in the LAN, by mapping port 8080 of the internal host 10.0.0.13 to port 80 of the external address 93.184.216.119

Redirect all outbound HTTP traffic originating from the LAN to a proxy running on port 3128 on the Linux box

sysctl -w net.ipv4.ip_forward=1

echo 1 > /proc/sys/net/ipv4/ip_forward

Enable IP forwarding; necessary to set up a Linux machine as a router. (This command causes other network options to be changed as well.)
In firewalld, a network interface (aka interface) or a subnet address (aka source) can be assigned to a specific zone. To determine to which zone a packet belongs, first the zone of the source is analyzed, then the zone of the interface; if no source or interface matches, the packet is associated to the default zone (which is "public", unless set otherwise). If the zone is not specified (via --zone=zone), the command is applied to the default zone. By default, commands are temporary; adding the --permanent option to a command sets it as permanent, or shows permanent settings only. Temporary commands are effective immediately but are canceled at reboot, firewall reload, or firewall restart. Permanent commands are effective only after reboot, firewall reload, or firewall restart.

<table>
<thead>
<tr>
<th>Firewalld zones</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>block</td>
<td>Rejects incoming connections with an ICMP HOST_PROHIBITED; allows only established connections</td>
</tr>
<tr>
<td>dmz</td>
<td>Used to expose services to the public; allows only specific incoming connections</td>
</tr>
<tr>
<td>drop</td>
<td>Drops all incoming packets; allows only outgoing connections</td>
</tr>
<tr>
<td>external</td>
<td>Used for routing and masquerading; allows only specific connections</td>
</tr>
<tr>
<td>home</td>
<td>Allows only specific incoming connections</td>
</tr>
<tr>
<td>internal</td>
<td>Used to define internal networks and allow only private network traffic</td>
</tr>
<tr>
<td>public</td>
<td>Allows only specific incoming connections. Default zone</td>
</tr>
<tr>
<td>trusted</td>
<td>Accepts all traffic</td>
</tr>
<tr>
<td>work</td>
<td>Used to define internal networks and allow only private network traffic</td>
</tr>
</tbody>
</table>

The list of firewall zones can be obtained via the command `firewall-cmd --get-zones`.

```
systemctl status firewalld
firewall-cmd --state
firewall-config
firewall-cmd --reload
firewall-cmd --complete-reload
firewall-cmd --runtime-to-permanent
firewall-cmd --list-all-zones
firewall-cmd --get-default-zone
firewall-cmd --set-default-zone=home
firewall-cmd --get-active-zones
firewall-cmd --get-zones
firewall-cmd --get-zone-of-interface=eth0
firewall-cmd --new-zone=test
firewall-cmd --zone=home --change-interface=eth0
firewall-cmd --zone=home --list-all
firewall-cmd --zone=home --list-all --permanent
firewall-cmd --zone=home --add-source=10.1.1.0/24
firewall-cmd --zone=home --list-sources
```

Check the status of the firewall
Firewall management GUI
Reload firewall configuration; this applies all permanent changes and cancels all temporary changes. Current connections are not terminated
Reload firewall configuration, stopping all current connections
Transform all temporary changes to permanent
List all zones and their full settings
Show the default zone
Set "home" as the default zone
Show the active zones i.e. zones bound to either an interface or a source
Show all available zones
Show the zone assigned to eth0
Create a new zone called "test"
Assign eth0 to the "home" zone
List temporary settings of the "home" zone
List permanent settings of the "home" zone
Assign 10.1.1.0/24 to the "home" zone i.e. route all traffic from that subnet to that zone
List sources bound to the "home" zone
firewall-cmd --zone=trusted --add-service=ssh
Add the SSH service to the "trusted" zone

firewall-cmd --zone=trusted --add-port=22/tcp
Add the SSH, HTTP, and HTTPS services to the "trusted" zone

firewall-cmd --zone=trusted --add-service={ssh,http,https}
Show temporary and permanent services bound to the "trusted" zone

firewall-cmd --zone=trusted --list-services
Show temporary and permanent ports open on the "trusted" zone

firewall-cmd --zone=trusted --list-ports
Add the SSH service to the "trusted" zone

firewall-cmd --get-services
List all predefined services

Predefined services are configured in /usr/lib/firewalld/services/service.xml.
User-defined services are configured in /etc/firewalld/services/service.xml.

firewall-cmd --get-icmptypes
Show all known types of ICMP messages

firewall-cmd --add-icmp-block=echo-reply
Block a specific ICMP message type

firewall-cmd --query-icmp-block=echo-reply
Tell if a specific ICMP message type is blocked

firewall-cmd --list-icmp-block
Show the list of blocked ICMP message types

firewall-cmd --add-rich-rule='richrule'
Set up a rich rule (for more complex and detailed firewall configurations)

firewall-cmd --add-rich-rule='rule family=ipv4 source address=10.2.2.0/24 service name=tftp log prefix=tftp level=info limit value=3/m accept'
Set up a rich rule to allow tftp connections from subnet 10.2.2.0/24 and log them via syslog at a rate of 3 per minute

firewall-cmd --list-rich-rules
List all rich rules

The manpage man firewalld.richlanguage contains several examples of rich rules.

firewall-cmd --direct --add-rule directrule
Set up a direct rule (in iptables format)

firewall-cmd --direct --add-rule \ ipv4 filter INPUT 0 -p tcp --dport 22 -j ACCEPT
Set up a direct rule to allow SSH connections

firewall-offline-cmd directrule
Set up a direct rule when firewalld is not running

firewall-cmd --direct --get-all-rules
Show all direct rules

User-defined direct rules are stored in /etc/firewalld/direct.xml.
The manpage man firewalld.direct documents the syntax of direct rules.

firewall-cmd --zone=zone --add-masquerade
Set up masquerading for hosts of zone; packets originating from zone will get the firewall's IP address on the "external" zone as source address

firewall-cmd --zone=zone --add-rich-rule='rule family=ipv4 source address=10.2.2.0/24 masquerade'
Set up masquerading only for those hosts of zone located in subnet 10.2.2.0/24

firewall-cmd --zone=zone --add-forward-port=\ port=22:proto=tcp:toport=2222:toaddr=10.7.7.7
Set up port forwarding for hosts of zone; incoming connections to port 22 for hosts of zone will be forwarded to port 2222 on host 10.7.7.7
Secure Shell (SSH) is a protocol (not a shell) for encrypted secure communications. It is mostly used as a replacement to Telnet to securely login to a remote server's terminal, but can be applied to any network protocol. Some of the most common applications of SSH are e.g. Secure Copy (SCP) and SSH File Transfer Protocol (SFTP).

**Secure Copy (SCP)**

Connect to a remote host via SSH and login as user.

```
ssh user@host
```

Options:

- `-v` `-vv` `-vvv` Increasing levels of verbosity
- `-p n` Use port `n` instead of standard port 22

**Command execution**

Execute a command on a remote host.

```
ssh user@host command
```

**Automatic connection monitoring**

Connect to a remote host, monitoring the connection and restarting it automatically if it dies.

```
autossh user@host
```

**Password login**

Connect to a remote host using the specified password.

```
sshpas -p password ssh user@host
```

**Execute a command in parallel on a group of remote hosts**

```
pssh -i -H "host1 host2 host3" command
```

**Generating keys**

Generate interactively a 2048-bit RSA key pair; will prompt for a passphrase.

```
ssh-keygen -t rsa -b 2048
```

Generate a DSA key pair.

```
ssh-keygen -t dsa
```

Change passphrase of the private key.

```
ssh-keygen -p -t rsa
```

Generate an RSA key with no passphrase (for non-interactive use) and no comment.

```
ssh-keygen -q -t rsa -f keyfile -N '' -C ''
```

View key length and fingerprint of a public or private key.

```
ssh-keygen -lf keyfile
```

View fingerprint of a key, calculated using `hashfunction`.

- RSA keys fingerprint use `sha1` (deprecated) or `md5`
- `sshash-keygen -t dsa`
- `sshash-keygen -p -t rsa`
- `sshash-keygen -q -t rsa -f keyfile -N '' -C ''`
- `ssh-keygen -lf keyfile`
- `< keyfile.pub awk '{print $2}' \ |
  base64 -d | openssl hashfunction`

**Get the public key of host and add it to the user’s known hosts file**

```
ssh-keyscan host >> ~/.ssh/known_hosts
```

**Echo to the terminal the environment variables that must be set in order to use the SSH Agent**

```
ssh-agent
```

```
eval `ssh-agent`
```

**Start ssh-agent and cache the specified key**

```
ssh-agent bash -c 'ssh-add keyfile'
```

**Add the default private keys to the ssh-agent cache**

```
ssh-add
```

**Add a specific private key to the ssh-agent cache**

```
ssh-add keyfile
```

**Use locally available keys to authorize, via public key authentication, login of user on a remote host.**

This is done by copying the user's local public key

```
~/.id_rsa.pub to ~/.ssh/authorized_keys on the remote host
```

```
ssh-copy-id user@host
```
scp /path1/file user@host:/path2/
scp user@host:/path1/file /path2/
scp user1@host1:/path1/file user2@host2:/path2/

Non-interactive secure file copy via SSH.
Can transfer files from local to remote, from remote to local,
or between two remote hosts

sftp user@host

SSH FTP-like tool for secure file transfer

scponly

SSH wrapper pseudo-shell providing access to remote users
for secure file transfer, but without execution privileges

sshfs user@host:/dir/ mountpoint/

SSH tool that allows mounting a remote directory as an SSH
filesystem on a mountpoint on the local machine. Uses the
FUSE kernel module.
The filesystem can be unmounted via the command
fusermount -u mountpoint/
SSH - operations

SSH port forwarding (aka SSH tunneling)

ssh -L 2525:mail.foo.com:25 user@mail.foo.com
Establish a SSH encrypted tunnel from localhost to remote host mail.foo.com, redirecting traffic from local port 2525 to port 25 of remote host mail.foo.com. Useful if the local firewall blocks outgoing port 25. In this case, port 2525 is used to go out; the application must be configured to connect to localhost on port 2525 (instead of mail.foo.com on port 25)

ssh -L 2525:mail.foo.com:25 user@login.foo.com
Establish a SSH encrypted tunnel from localhost to remote host login.foo.com. Remote host login.foo.com will then forward, unencrypted, all data received over the tunnel on port 2525 to remote host mail.foo.com on port 25

SSH reverse forwarding (aka SSH reverse tunneling)

ssh -R 2222:localhost:22 user@login.foo.com
Establish a SSH encrypted reverse tunnel from remote host login.foo.com back to localhost, redirecting traffic sent to port 2222 of remote host login.foo.com back towards local port 22. Useful if the local firewall blocks incoming connections so remote hosts cannot connect back to local machine. In this case, port 2222 of login.foo.com is opened for listening and connecting back to localhost on port 22; remote host login.foo.com is then able to connect to the local machine on port 2222 (redirected to local port 22)

SSH as a SOCKS proxy

ssh -D 33333 user@login.foo.com
The application supporting SOCKS must be configured to connect to localhost on port 33333. Data is tunneled from localhost to login.foo.com, then unencrypted to destination

X11 Forwarding

ssh -X user@login.foo.com
Enable the local display to execute locally an X application stored on a remote host login.foo.com

How to enable public key authentication

1. On remote host, set PubkeyAuthentication yes in /etc/ssh/sshd_config
2. On local machine, do ssh-copy-id you@remotehost (or copy your public key to the remote host by hand)

How to enable host-based authentication amongst a group of trusted hosts

1. On all hosts, set HostbasedAuthentication yes in /etc/ssh/sshd_config
2. On all hosts, create /etc/ssh/hosts.equiv and enter in this file all trusted hostnames
3. Connect via SSH manually from your machine on each host so that all hosts’ public keys go into ~/.ssh/known_hosts
4. Copy ~/.ssh/known_hosts from your machine to /etc/ssh/ssh_known_hosts on all hosts

How to enable X11 Forwarding

1. On remote host 10.2.2.2, set X11Forwarding yes in /etc/ssh/sshd_config, and make sure that xauth is installed
2. On local host 10.1.1.1, type ssh -X 10.2.2.2, then run on remote host the graphical application e.g. xclock &

It is also possible to enable X11 Forwarding via telnet (unencrypted, therefore insecure and not recommended):

1. On remote host 10.2.2.2, type export DISPLAY=10.1.1.1:0.0
2. On local host 10.1.1.1, type xhost +
3. On local host 10.1.1.1, type telnet 10.2.2.2, then run on remote host the graphical application e.g. xclock &
### /etc/ssh/sshd_config

**SSH server configuration file**

- **PermitRootLogin yes**
  - Control superuser login via SSH. Possible values are:
    - yes: Superuser can login
    - no: Superuser cannot login
    - without-password: Superuser cannot login with password
    - forced-commands-only: Superuser can only run commands in SSH command line

- **AllowUsers jdoe ksmith**
  - List of users that can/cannot login via SSH, or * for everybody

- **DenyUsers jhacker**

- **AllowGroups geeks**
  - List of groups whose members can/cannot login via SSH, or * for all groups

- **DenyGroups ***

- **PasswordAuthentication yes**
  - Permit authentication via login and password

- **PubKeyAuthentication yes**
  - Permit authentication via public key

- **HostbasedAuthentication yes**
  - Permit authentication based on trusted hosts

- **Protocol 1,2**
  - Specify protocols supported by SSH. Value can be 1 or 2 or both

- **X11Forwarding yes**
  - Allow X11 Forwarding

### /etc/ssh/ssh_config and ~/.ssh/config

**SSH client configuration file**

- **Host ***
  - List of hosts to which the following directives will apply, or * for all hosts

- **StrictHostKeyChecking yes**
  - Ask before adding new host keys to the ~/.ssh/known_hosts file, and refuse to connect if the key for a known host has changed. This prevents MITM attacks

- **GSSAPIAuthentication yes**
  - Support authentication using GSSAPI

- **ForwardX11Trusted yes**
  - Allow remote X11 clients to fully access the original X11 display

- **IdentityFile ~/.ssh/id_rsa**
  - User identity file for authentication. Default values are:
    - ~/.ssh/identity for protocol version 1
    - ~/.ssh/id_rsa and ~/.ssh/id_dsa for protocol version 2
The X.509 standard defines the format of public key certificates and other related files. It includes cryptographic standards and protocols such as SSL/TLS, PKCS7, PKCS12, and OCSP.

The Public Key Infrastructure X.509 (PKIX) is described in RFC 5280.

### X.509 file formats

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DER</td>
<td>Binary-encoded certificate</td>
</tr>
</tbody>
</table>
| PEM       | ASCII-armored Base64-encoded certificate, included between these two lines: 

```
-----BEGIN FILE_TYPE-----
-----END FILE_TYPE-----
```

where *FILE_TYPE* is one of the X.509 file types (see below)

DER and PEM are also used as file extensions for different types of files (see below).

### X.509 file type extensions

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRT</td>
<td>Certificate or certificate chain</td>
</tr>
<tr>
<td>CER</td>
<td>Certificate or certificate chain</td>
</tr>
<tr>
<td>CSR</td>
<td>Certificate Signing Request</td>
</tr>
<tr>
<td>KEY</td>
<td>Private key</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>DER</td>
<td>Certificate; DER-encoded</td>
</tr>
<tr>
<td>PEM</td>
<td>Certificate (including or not the private key), certificate chain, or Certificate Signing Request; PEM-encoded</td>
</tr>
</tbody>
</table>

### Other file type extensions

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P12</td>
<td>Certificate (including or not the private key), certificate chain, or Certificate Signing Request; bundled in a PKCS#12 archive file format</td>
</tr>
<tr>
<td>PFX</td>
<td>Certificate (including or not the private key), certificate chain, or Certificate Signing Request; bundled in a PKCS#12 archive file format</td>
</tr>
</tbody>
</table>
OpenSSL is an open source cryptographic library containing an implementation of the SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols, plus various cryptographic utilities.

**openssl**  
OpenSSL command line tool

**genkey**  
Interactive utility for the generation of SSL certificates and Certificate Signing Requests

**certwatch**  
Program that issues email warnings when an SSL certificate is close to its expiration date

**keyrand**  
Utility that collects random bits from `/dev/random` and appends them to a file

---

**CA.pl**  
User-friendly command for common certificate operations

**CA.pl -newca**  
Create a Certification Authority hierarchy

**CA.pl -newreq**  
Generate a Certificate Signing Request

**CA.pl -newreq-nodes**  
Generate a Certificate Signing Request, creating also a key pair (unencrypted, for non-interactive use)

**CA.pl -signreq**  
Sign a Certificate Signing Request

**CA.pl -pkcs12 "Certificate name"**  
Generate a PKCS#12 certificate from a Certificate Signing Request

**CA.pl -newcert**  
Generate a self-signed certificate

**CA.pl -verify**  
Verify a certificate against the Certification Authority certificate for "demoCA"
OpenSSL - commands

openssl x509 -text -in cert.crt -noout
openssl req -text -in cert.csr -noout
openssl req -new -key cert.key -out cert.csr
openssl req -new -keyout cert.key -out cert.csr \
-newkey rsa:2048 -nodes
openssl x509 -req -in cert.csr -CAcreateserial \ 
-CA ca.crt -CAkey ca.key -out cert.crt -days validity
openssl req -x509 -keyout cert.key -out cert.crt \
-newkey rsa:2048 -nodes -days validity
openssl ca -config ca.conf -in cert.crt \ 
-out cert.crt -days validity -verbose
openssl ca -config ca.conf -generl -revoke cert.crt \
-crl_reason why
openssl ca -config ca.conf -generl -out list.crl

openssl x509 -in cert.pem -outform DER -out cert.der
openssl pkcs12 -export -in cert.pem \ 
inkey cert.key -out cert.pfx -name friendlyname
openssl pkcs12 -in cert.p12 -out cert.crt -clcerts \ 
-nokeys
openssl pkcs12 -in cert.p12 -out cert.key -nocerts \ 
-nodes
openssl pkcs12 -in cert.p12 -out ca.crt -cacerts
cat cert.crt cert.key > cert.pem

openssl dgst -hashfunction -out file.hash file
openssl dgst -hashfunction file | cmp -b file.hash
openssl dgst -hashfunction -sign private.key \ 
-out file.sig file
openssl dgst -hashfunction -verify public.key \ 
-signature file.sig file
openssl enc -e -cipher -in file -out file.enc -salt
openssl enc -d -cipher -in file.enc -out file

openssl genpkey -algorithm RSA -cipher 3des \ 
-pkeyopt rsa_keygen_bits:2048 -out keypair.pem
openssl pkey -text -in private.key -noout
openssl pkey -in old.key -out new.key -cipher
openssl pkey -in old.key -out new.key

1. openssl s_client -connect www.site.com:443 > tmpfile
2. CTRL-C
3. openssl x509 -in tmpfile -text

openssl list-message-digest-commands
openssl list-cipher-commands

Read a certificate
Read a Certificate Signing Request
Generate a Certificate Signing Request, given a private key
Generate a Certificate Signing Request, creating also a 2048-bit RSA key pair (unencrypted, for non-
interactive use)
Sign a certificate as a CA, given a Certificate Signing Request
Generate a self-signed root certificate, and create a new CA private key
Sign a certificate
Revoke a certificate
Generate a Certificate Revocation List containing all revoked certificates so far
Convert a certificate from PEM to DER
Convert a certificate from PEM to PKCS#12 including the private key
Convert a certificate from PKCS#12 to PEM
Extract the private key from a PKCS#12 certificate
Create a PEM certificate from CRT and private key
Generate the digest (hash) of a file
Check the hash of a file; no output means OK
Sign a file
Verify the signature of a file
Encrypt a file
Decrypt a file
Generate a 2048-bit RSA key pair protected by a TripleDES-encrypted passphrase
Examine a private key
Change the passphrase of a private key
Remove the passphrase from a private key
Inspect an SSL certificate from a website
List all available hash functions
List all available ciphers
In **symmetric cryptography**, a symmetric cipher (i.e. cryptographic algorithm) is used with a *shared secret key* to encrypt a message. The message can then be decrypted using the same key.

In **asymmetric cryptography** aka **Public Key Cryptography**, ciphers do not operate with a single key but with a *key pair*, composed of a *public key* and a *private key*. Public and private key are created together at the same time using a special algorithm and are strictly related to each other; however, deriving a private key from its public key is computationally infeasible. A message is encrypted with a public key and can only be decrypted with the companion private key. Similarly, a message is digitally signed with a private key and can only be verified with the companion public key.

Encryption guarantees confidentiality (only sender and receiver are able to know the contents of the message). Digital signature guarantees authentication (the receiver can verify that the message originates from the intended sender), integrity (the receiver can verify that the message was not modified since it was signed), and non-repudiation (the sender cannot deny having signed the message).

In a **block cipher**, the input plaintext is split into blocks of fixed size, fed to the algorithm, and transformed according to the key. If the input plaintext is shorter than the block length, padding is used.

In a **stream cipher**, the input plaintext is combined with a pseudorandom digit stream. The key is applied to each bit, one at a time.

A **hash function** maps a bitstring of arbitrary length to another bitstring of fixed length, hence outputting a condensed representative image of the bitstring fed in input. Changing just one bit of the input string results in a very different hash value in output (avalanche effect).

A hash function must have the following properties:
- be one-way, i.e. given an output value it is computationally infeasible to find the matching input (preimage resistance)
- given a specific input, it is computationally infeasible to find another input that results in the same output (second preimage resistance)
- it is computationally infeasible to find two different inputs which would result in the same output (collision resistance)

A **Public Key Infrastructure (PKI)** handles creation, management, distribution, use, and revocation of Digital Certificates. It is composed of the following entities:
- **Certification Authority (CA)**, which issues and verifies Digital Certificates
- **Registration Authority (RA)**, which verifies the CA, verifies the subject, and ensures valid and correct registration
- **Validation Authority (VA)**, which verifies the validity of a Digital Certificate

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcrypt</td>
<td>File encryption tool. Uses the Blowfish algorithm</td>
</tr>
<tr>
<td>ccrypt</td>
<td>File and stream encryption tool. Uses the Rijndael block cipher</td>
</tr>
<tr>
<td>ccr</td>
<td>Codecrypt, an encryption and signing tool that uses only algorithms resistant to quantum-computer cryptanalysis</td>
</tr>
<tr>
<td>age</td>
<td>File encryption tool</td>
</tr>
<tr>
<td>stegsnow</td>
<td>Steganography tool for text files. The secret message is concealed in additional tab and whitespace characters at the end of lines</td>
</tr>
<tr>
<td>steghide</td>
<td>Steganography tool for image and audio files</td>
</tr>
<tr>
<td>shasum</td>
<td>Print or check the digest of a file generated by the SHA hashing algorithm</td>
</tr>
<tr>
<td>sha1sum</td>
<td></td>
</tr>
<tr>
<td>sha224sum</td>
<td></td>
</tr>
<tr>
<td>sha256sum</td>
<td></td>
</tr>
<tr>
<td>sha384sum</td>
<td></td>
</tr>
<tr>
<td>sha512sum</td>
<td></td>
</tr>
<tr>
<td>md5sum</td>
<td>Print or check the digest of a file generated by the MDS hashing algorithm</td>
</tr>
</tbody>
</table>
### Symmetric ciphers

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES (Data Encryption Standard)</td>
<td>Block cipher with a 64-bit block size. Uses DEA (Data Encryption Algorithm) with a 56-bit key. Obsolete and insecure.</td>
</tr>
</tbody>
</table>
| 3DES (Triple DES) | Cipher which uses a key bundle of three DES keys: \(K_1\), \(K_2\), and \(K_3\). The algorithm consists in applying DES three times: encrypt with \(K_1\), decrypt with \(K_2\), then encrypt with \(K_3\). Key options are:  
- \(K_1\), \(K_2\), and \(K_3\) are all different (most secure)  
- \(K_1 = K_3\)  
- \(K_1 = K_2 = K_3\) (fallback to DES, insecure) |
| IDEA (International Data Encryption Algorithm) | Block cipher with a 64-bit block size and a 128-bit key. Intended as a replacement for DES, is now patent-free. |
| RC4 (Rivest Cipher 4) aka ARC4 or ARCFOUR | Stream cipher with byte-oriented operations, based on the use of a random permutation. Key size is variable. Patented by RSA Security. |
| RC5 (Rivest Cipher 5) | Fast, parameterized block cipher with a 32-bit, 64-bit, or 128-bit block size. Number of rounds can be up to 255, and key size can be up to 2040 bits. Uses two 2-bit registers. Patented by RSA Security. |
| RC6 (Rivest Cipher 6) | Block cipher derived from RC5, with the improvement of integer multiplication and four 4-bit registers. Patented by RSA Security. |
| SEAL (Software-optimized Encryption Algorithm) | Stream cipher optimized for 32-bit machines. Patented by IBM. |
| Blowfish | 16-round Feistel block cipher with a 64-bit block size. Key size can be from 32 to 448 bits, and is expanded into a set of subkeys 4168 bytes long. Designed by Bruce Schneier as a replacement for DES or IDEA. |
| Twofish | Feistel block cipher with a 128-bit block size. Highly configurable. Designed by Bruce Schneier et al. |

### Asymmetric ciphers

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSA (Digital Signature Algorithm)</td>
<td>NIST standard used in the Digital Signature Standard (DSS). Generates a 320-bit digital signature. Based on the algebraic properties of modular exponentiation and the discrete logarithm problem.</td>
</tr>
<tr>
<td>RSA (Rivest-Shamir-Adleman)</td>
<td>Cryptosystem for encryption and authentication. Based on modular arithmetic and large prime numbers.</td>
</tr>
</tbody>
</table>
| DH (Diffie-Hellman) | Cryptographic protocol for securely establishing a shared secret key over an insecure channel. Based on modular arithmetic, exponentiation, and large prime numbers. Strength and algorithm of key exchange are defined by groups:  
- group 1 (768 bit)  
- group 2 (1024 bit)  
- group 5 (1536 bit)  
- group 14 (2048 bit)  
- group 19 (256 bit, elliptic curve)  
- group 20 (384 bit, elliptic curve) |
### Hash functions

<table>
<thead>
<tr>
<th>Hash function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MD2</strong>&lt;br&gt;(Message Digest v2)</td>
<td>Takes in input a message which is a multiple of 512 bits (if not, padding is used) and has a maximum length of $2^{512}-1$ bits, and generates a 128-bit hash. Supports 8-bit machines (i.e. word size is 8 bits). Broken, but still used in digital certificates together with RSA.</td>
</tr>
<tr>
<td><strong>MD4</strong>&lt;br&gt;(Message Digest v4)</td>
<td>Same properties as MD2. Supports 32-bit machines. Proven severely insecure, thus obsolete.</td>
</tr>
<tr>
<td><strong>MD5</strong>&lt;br&gt;(Message Digest v5)</td>
<td>Same properties as MD2. Supports 32-bit machines. This is the hash function of the MD family currently in use.</td>
</tr>
<tr>
<td><strong>SHA-0</strong>&lt;br&gt;(Secure Hash Algorithm v0)</td>
<td>Original version of the SHA hash function. Vulnerable, thus not in use anymore.</td>
</tr>
<tr>
<td><strong>SHA-1</strong>&lt;br&gt;(Secure Hash Algorithm v1)</td>
<td>Takes in input a message with a maximum length of $2^{64}-1$ bits, and generates a 160-bit hash. Vulnerable and therefore no longer approved for cryptographic use.</td>
</tr>
<tr>
<td><strong>SHA-2</strong>&lt;br&gt;(Secure Hash Algorithm v2)</td>
<td>Family of hash functions:&lt;br&gt;- <strong>SHA-256</strong> (for 32-bit machines, generates a 256-bit hash)&lt;br&gt;- <strong>SHA-224</strong> (for 32-bit machines, generates a 224-bit hash, truncated version)&lt;br&gt;- <strong>SHA-512</strong> (for 64-bit machines, generates a 512-bit hash)&lt;br&gt;- <strong>SHA-384</strong> (for 64-bit machines, generates a 384-bit hash, truncated version)&lt;br&gt;Max input message length is $2^{256}-1$ bits.</td>
</tr>
<tr>
<td><strong>SHA-3</strong>&lt;br&gt;(Secure Hash Algorithm v3)</td>
<td>Family of hash functions:&lt;br&gt;- <strong>SHA3-256</strong> (for 32-bit machines, generates a 256-bit hash)&lt;br&gt;- <strong>SHA3-224</strong> (for 32-bit machines, generates a 224-bit hash, truncated version)&lt;br&gt;- <strong>SHA3-512</strong> (for 64-bit machines, generates a 512-bit hash)&lt;br&gt;- <strong>SHA3-384</strong> (for 64-bit machines, generates a 384-bit hash, truncated version)&lt;br&gt;- <strong>SHAKE128</strong> (generates a hash of arbitrary length)&lt;br&gt;- <strong>SHAKE256</strong> (generates a hash of arbitrary length)&lt;br&gt;Max input message length is unlimited.</td>
</tr>
<tr>
<td><strong>RIPEMD</strong>&lt;br&gt;(RACE Integrity Primitives Evaluation Message Digest)</td>
<td>Family of hash functions:&lt;br&gt;- <strong>RIPEMD</strong> (generates a 128-bit hash; insecure)&lt;br&gt;- <strong>RIPEMD-128</strong> (generates a 128-bit hash; insecure)&lt;br&gt;- <strong>RIPEMD-160</strong> (generates a 160-bit hash; most commonly used)&lt;br&gt;- <strong>RIPEMD-256</strong> (generates a 256-bit hash)&lt;br&gt;- <strong>RIPEMD-320</strong> (generates a 320-bit hash)</td>
</tr>
</tbody>
</table>

### Authentication systems

<table>
<thead>
<tr>
<th>Authentication system</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMAC</strong>&lt;br&gt;(Hash-based Message Authentication Code)</td>
<td>Message Authentication Code used to verify data integrity and sender authentication. Uses a hash function in conjunction with a secret key.</td>
</tr>
<tr>
<td><strong>PAP</strong>&lt;br&gt;(Password Authentication Protocol)</td>
<td>Password-based authentication protocol used by Point-to-Point Protocol (PPP) to authenticate remote clients. Uses a weak authentication scheme, vulnerable to attacks; passwords are transmitted in plaintext over the network. For this reason, is not recommended and it is advised to use CHAP or EAP instead.</td>
</tr>
<tr>
<td><strong>CHAP</strong>&lt;br&gt;(Challenge-Handshake Authentication Protocol)</td>
<td>Authentication protocol used by Point-to-Point Protocol (PPP) to authenticate remote clients. Client identity is verified via a three-way handshake. It uses an incrementally changing identifier and a variable challenge value in order to thwart replay attacks.</td>
</tr>
<tr>
<td><strong>EAP</strong>&lt;br&gt;(Extensible Authentication Protocol)</td>
<td>Authentication framework able to use different authentication systems (passwords, smart tokens, one-time passwords, Secure ID cards, digital certificates, public key cryptography protocols, etc.) via a challenge-response mechanism.</td>
</tr>
<tr>
<td><strong>LEAP</strong>&lt;br&gt;(Lightweight Extensible Authentication Protocol)</td>
<td>Cisco proprietary version of EAP, used for WEP. Uses either the MS-CHAP or the EAP-FAST authentication protocol. Vulnerable and not recommended.</td>
</tr>
<tr>
<td><strong>PEAP</strong>&lt;br&gt;(Protected Extensible Authentication Protocol)</td>
<td>TLS-encapsulated secured version of EAP, used in WPA2.</td>
</tr>
</tbody>
</table>
GPG (GNU Privacy Guard) aka GnuPG is a well-known implementation of the OpenPGP standard described in RFC 4880. The OpenPGP standard derives from PGP (Pretty Good Privacy), the first tool for strong encryption available to the public. It specifies a suite of algorithms: ElGamal, DSA, Triple DES, SHA-1, RSA, AES-128, CAST-128, IDEA, Camellia, ECC (Elliptic Curve Cryptography) i.e. ECDSA and ECDH.

```
gpg --gen-key

Generate a key pair

Example: gpg --gen-key

Import Alice's public key alice.asc into your keyring

```
gpg --import alice.asc

```
gpg --list-keys

List the keys contained into your keyring

```
Example: gpg --list-keys

List your private keys contained into your keyring

```
gpg --list-secret-keys

Example: gpg --list-secret-keys

List the public keys contained into your keyring

```
gpg --list-public-keys

Example: gpg --list-public-keys

Export your whole keyring to a file keyring.gpg

```
gpg --export -o keyring.gpg

Export your private key to a file private.key

```
gpg --export-secret-key -a "You" -o private.key

Export Alice's public key to a file alice.pub

```
gpg --export-public-key -a "Alice" -o alice.pub

Sign Alice's public key

```
gpg --edit-key "Alice"

Sign file (with your private key) and encrypt it to Alice (with Alice's public key)

```
gpg -e -u "You" -r "Alice" file

Decrypt file.gpg (with your own private key) and save the decrypted file to file

```
gpg -d file.gpg -o file
```
LUKS (Linux Unified Key Setup) is a platform-independent specification for the encryption of a block device. It uses dm-crypt, a transparent disk encryption subsystem which is part of the device mapper, as a backend.

**cryptsetup**

Frontend command for dm-crypt. Will prompt for a passphrase for most operations on a LUKS-encrypted device

**cryptsetup luksFormat device**

Initialize a LUKS partition, prompting for an encryption passphrase

**cryptsetup luksChangeKey device**

Change the passphrase of a LUKS partition

**cryptsetup luksAddKey device**

Add a new passphrase to a LUKS partition

**cryptsetup luksAddKey device keyfile**

Add a new keyfile to a LUKS partition

**cryptsetup luksRemoveKey device**

Remove a passphrase from a LUKS partition

**cryptsetup luksRemoveKey device keyfile**

Remove a keyfile from a LUKS partition

**cryptsetup luksKillSlot device keyslot**

Remove a key from a LUKS partition

**cryptsetup isLuks device**

Return true if the device is a LUKS partition

**cryptsetup luksDump device**

Dump the header information of a LUKS partition

**cryptsetup luksUUID device**

Print the UUID a LUKS partition

**cryptsetup luksOpen device name**

Open a LUKS device and set up a mapping name

**cryptsetup luksClose name**

Close a LUKS device and remove the mapping name

**cryptsetup luksSuspend name**

Suspend a LUKS device and wipe the encryption key from memory

**cryptsetup luksResume name**

Resume a suspended LUKS device

**cryptsetup luksHeaderBackup device --header-backup-file file**

Backup header and keyslot areas of a LUKS device to a file

**cryptsetup luksHeaderRestore device --header-backup-file file**

Restore header and keyslot areas of a LUKS device from a file
OpenVPN is an open source software that implements a Virtual Private Network (VPN) between two endpoints. The encrypted VPN tunnel uses UDP port 1194.

openvpn --genkey --secret keyfile

Generate a shared secret keyfile for OpenVPN authentication. The keyfile must be copied on both server and client

openvpn server.conf

Start the VPN on the server side

openvpn client.conf

Start the VPN on the client side

/etc/openvpn/server.conf

Server-side configuration file:

dev tun
ifconfig server_IP client_IP
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
secret keyfile

/etc/openvpn/client.conf

Client-side configuration file:

remote server_public_IP
dev tun
ifconfig client_IP server_IP
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
secret keyfile
<table>
<thead>
<tr>
<th>Key</th>
<th>Alternate key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{CTRL} + \texttt{F}</td>
<td></td>
<td>Move cursor forward one character</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{B}</td>
<td></td>
<td>Move cursor backward one character</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{A}</td>
<td>\texttt{HOME}</td>
<td>Move cursor to beginning of line</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{E}</td>
<td>\texttt{END}</td>
<td>Move cursor to end of line</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{H}</td>
<td>\texttt{BACKSPACE}</td>
<td>Delete character to the left of cursor</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{W}</td>
<td></td>
<td>Delete word to the left of cursor</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{U}</td>
<td></td>
<td>Delete all characters to the left of cursor</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{K}</td>
<td></td>
<td>Delete all characters to the right of cursor</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{T}</td>
<td>\texttt{ESC} + \texttt{T}</td>
<td>Swap current character with previous one</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{U}</td>
<td>\texttt{ESC} + \texttt{U}</td>
<td>Swap current word with previous one</td>
</tr>
<tr>
<td>\texttt{SHIFT} + \texttt{PAGE UP}</td>
<td>\texttt{↑}</td>
<td>Scroll up the screen buffer</td>
</tr>
<tr>
<td>\texttt{SHIFT} + \texttt{PAGE DOWN}</td>
<td>\texttt{↓}</td>
<td>Scroll down the screen buffer</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{L}</td>
<td>\texttt{CTRL} + \texttt{L}</td>
<td>Clear screen (same as \texttt{clear})</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{P}</td>
<td>\texttt{↑}</td>
<td>Previous command in history</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{N}</td>
<td>\texttt{↓}</td>
<td>Next command in history</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{R}</td>
<td>\texttt{ALT} + \texttt{/}</td>
<td>Reverse history search</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{I}</td>
<td>\texttt{TAB}</td>
<td>Autocomplete commands, filenames, and directory names</td>
</tr>
<tr>
<td>\texttt{ALT} + \texttt{J}</td>
<td>\texttt{ALT} + \texttt{J}</td>
<td>Autocomplete filenames and directory names only</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{ALT} + \texttt{L}</td>
<td>\texttt{ALT} + \texttt{L}</td>
<td>Expand the Bash alias currently entered on the command line</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{J}</td>
<td>\texttt{RETURN}</td>
<td>Line feed</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{M}</td>
<td>\texttt{RETURN}</td>
<td>Carriage return</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{S}</td>
<td>\texttt{ALT} + \texttt{F1}</td>
<td>Pause transfer to terminal</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{Q}</td>
<td>\texttt{CTRL} + \texttt{S}</td>
<td>Forward history search (if XON/XOFF flow control is disabled)</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{Z}</td>
<td>\texttt{CTRL} + \texttt{Q}</td>
<td>Resume transfer to terminal</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{C}</td>
<td>\texttt{CTRL} + \texttt{Z}</td>
<td>Send a SIGTSTP to put the current job in background</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{D}</td>
<td>\texttt{CTRL} + \texttt{D}</td>
<td>Send a SIGINT to stop the current process</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{ALT} + \texttt{DEL}</td>
<td>\texttt{CTRL} + \texttt{ALT} + \texttt{DEL}</td>
<td>Send a EOF to current process (same as \texttt{logout} if process is a shell)</td>
</tr>
<tr>
<td>\texttt{CTRL} + \texttt{ALT} + \text{F1 ... F6}</td>
<td>\texttt{CTRL} + \texttt{ALT} + \text{F1 ... F6}</td>
<td>Switch between text consoles (same as \texttt{chvt n})</td>
</tr>
<tr>
<td>Key</td>
<td>Alternate key</td>
<td>Function</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>CTRL ALT F7</td>
<td></td>
<td>Switch between X Window consoles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Increase X Window screen resolution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decrease X Window screen resolution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Switch between X Window tasks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Switch to next workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Switch to previous workspace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reboot the X Window server</td>
</tr>
</tbody>
</table>

### GNOME

<table>
<thead>
<tr>
<th>Key</th>
<th>Alternate key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT TAB</td>
<td></td>
<td>Switch between windows in the current workspace</td>
</tr>
<tr>
<td>SUPER</td>
<td></td>
<td>Show activities overview</td>
</tr>
<tr>
<td>SUPER -</td>
<td></td>
<td>Lock screen</td>
</tr>
<tr>
<td>SUPER</td>
<td></td>
<td>Show tray messages</td>
</tr>
<tr>
<td>SUPER -</td>
<td></td>
<td>Maximize current window</td>
</tr>
<tr>
<td>SUPER -</td>
<td></td>
<td>Restore normal size of current window</td>
</tr>
<tr>
<td>SUPER -</td>
<td></td>
<td>Maximize current window to left half screen</td>
</tr>
<tr>
<td>SUPER -</td>
<td></td>
<td>Maximize current window to right half screen</td>
</tr>
<tr>
<td>ALT F2</td>
<td></td>
<td>Run command</td>
</tr>
<tr>
<td>CTRL -</td>
<td></td>
<td>Increase terminal font size</td>
</tr>
<tr>
<td>CTRL -</td>
<td></td>
<td>Decrease terminal font size</td>
</tr>
</tbody>
</table>
The Hardware Abstraction Layer (HAL) manages device files and provides plug-and-play facilities. The HAL daemon hald maintains a persistent database of devices.
udev is the device manager for the Linux kernel. It dynamically generates the device nodes in /dev/ for devices present on the system; it also provides persistent naming for storage devices in /dev/disk.
When a device is added, removed, or changes state, the kernel sends an uevent received by the udevd daemon which will pass the uevent through a set of rules stored in /etc/udev/rules.d/*.rules and /lib/udev/rules.d/*.rules.

```
udevadm monitor
udevmonitor
udevadm info --attribute-walk --name=/dev/sda
cat /sys/block/sda/size
udevadm test /dev/sdb
gnome-device-manager
```

Show all kernel uevents and udev messages
Print all attributes of device /dev/sda in udev rules key format
Print the size attribute of disk sda in 512-byte blocks.
This information is retrieved from sysfs
Simulate an udev event run for the device and print debug output
Browser for the HAL device manager

---

```
/etc/udev/rules.d/*.rules and /lib/udev/rules.d/*.rules udev rules
```

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KERNEL==&quot;hda&quot;, NAME=&quot;mydisk&quot;</td>
<td>Match a device which was named by the kernel as hda; name the device node as &quot;mydisk&quot;. The device node will be therefore /dev/mydisk</td>
</tr>
<tr>
<td>KERNEL==&quot;hdb&quot;, DRIVER==&quot;ide-disk&quot;, SYMLINK=&quot;mydisk myhd&quot;</td>
<td>Match a device with kernel name and driver as specified; name the device node with the default name and create two symbolic links /dev/mydisk and /dev/myhd pointing to /dev/hdb</td>
</tr>
<tr>
<td>KERNEL==&quot;fd[0-9]*&quot;, NAME=&quot;floppy/%n&quot;, SYMLINK=&quot;%k&quot;</td>
<td>Match all floppy disk drives (i.e. fdn); place device node in /dev/floppy/n and create a symlink /dev/fd\n to it</td>
</tr>
<tr>
<td>SUBSYSTEM==&quot;block&quot;, ATTR{size}==&quot;41943040&quot;, SYMLINK=&quot;mydisk&quot;</td>
<td>Match a block device with a size attribute of 41943040; create a symlink /dev/mydisk</td>
</tr>
<tr>
<td>KERNEL==&quot;fd[0-9]*&quot;, OWNER=&quot;jdoe&quot;</td>
<td>Match all floppy disk drives; give ownership of the device file to user “jdoe”</td>
</tr>
<tr>
<td>KERNEL==&quot;sda&quot;, PROGRAM=&quot;/bin/mydevicenamer %k&quot;, SYMLINK=&quot;%c&quot;</td>
<td>Match a device named by the kernel as sda; to name the device, use the defined program which takes on stdin the kernel name and output on stdout e.g. name1 name2. Create symlinks /dev/name1 and /dev/name2 pointing to /dev/sda</td>
</tr>
<tr>
<td>KERNEL==&quot;sda&quot;, ACTION==&quot;add&quot;, RUN=&quot;/bin/myprogram&quot;</td>
<td>Match a device named by the kernel as sda; run the defined program when the device is connected</td>
</tr>
<tr>
<td>KERNEL==&quot;sda&quot;, ACTION==&quot;remove&quot;, RUN=&quot;/bin/myprogram&quot;</td>
<td>Match a device named by the kernel as sda; run the defined program when the device is disconnected</td>
</tr>
</tbody>
</table>

%\n = kernel number (e.g. = 3 for fd3)
%\k = kernel name (e.g. = fd3 for fd3)
%\c = device name as output from program

---

© Daniele Raffo
dr0.ch
A kernel version number has the form `major.minor.patchlevel`.

Kernel images are usually gzip-compressed and can be of two types: zImage (max 520 Kb) and bzImage (no size limit). Kernel modules can be loaded dynamically into the kernel to provide additional functionalities on demand, instead of being included when the kernel is compiled; this reduces memory footprint. `kerneld` (daemon) and `kmod` (kernel thread) facilitate the dynamic loading of kernel modules.

```
/lib/modules/X.Y.Z/*.ko
/lib/modules/X.Y.Z/modules.dep
```

Kernel modules for kernel version `X.Y.Z`

Modules dependencies. This file needs to be recreated (via the command `depmod -a`) after a reboot or a change in module dependencies.

```
/etc/modules.conf
/etc/conf.modules (deprecated)
```

Modules configuration file

```
/usr/src/linux/
/usr/src/linux/.config
```

Directory containing the kernel source code to be compiled

Kernel configuration file

```
/etc/initramfs-tools/initramfs.conf (Debian)
```

Configuration file for the initrd image file

```
freeramdisk
```

Free the memory used for the initrd image. This command must be run directly after unmounting `/initrd`

```
mkinitrd initrd_image kernel_version (Red Hat)
mkinitramfs (Debian)
dracut
```

Create an initrd image file

Create initial ramdisk images for preloading modules

```
dbus-monitor
dbus-monitor --session
dbus-monitor --system
```

Monitor messages going through a D-Bus message bus

Monitor session messages (default)

Monitor system messages

```
 LD_LIBRARY_PATH
/etc/ld.so.cache
/lib and /usr/lib
```

Shared library locations (other than the default ones /lib and /usr/lib) can be specified in the file `/etc/ld.so.conf`.

```
ldconfig
```

Create a cache file `/etc/ld.so.cache` of all available dynamically linked libraries. This command should be run when the system complains about missing libraries

```
ldd program_or_lib
```

Print library dependencies

---

The runtime loader `ld.so` loads the required shared libraries of the program into RAM, searching in this order:

1. `LD_LIBRARY_PATH` Environment variable specifying the list of dirs where libraries should be searched for first
2. `/etc/ld.so.cache` Cache file
3. `/lib` and `/usr/lib` Default locations for shared libraries

---
Kernel management

**lspci**
List PCI devices

**lspci -d 8086:**
List all Intel hardware present. PCI IDs are stored in:

- /usr/share/hwdata/pci.ids (Red Hat)
- /usr/share/misc/pci.ids (Debian)

**lsusb**
List USB devices

**lsusb -d 8086:**
List all Intel USB devices present. USB IDs are stored in:

- /usr/share/hwdata/usb.ids (Red Hat)
- /var/lib/usbutils/usb.ids (Debian)

**lsdev**
List information about the system hardware

**lshw**
List system hardware

**lscpu**
List information about the CPU architecture

**uname**
Print system information. Options to show the relevant piece of information are:

- `s` Kernel name
- `n` Network node hostname
- `r` Kernel release number X.Y.Z
- `v` Kernel version number
- `m` Machine hardware name
- `p` Processor type
- `i` Hardware platform
- `o` Operating system
- `a` All the above information, in that order

**evtest**
Monitor and query input device events in /dev/input/eventn

**dmesg**
Print the messages of the kernel ring buffer. Each entry is prepended by a timestamp showing the number of seconds since the machine booted up. Options are:

- `-T` Print human-readable timestamps
- `-n 1` Set the logging level to 1 (i.e. only panic messages)

**journalctl**
Display the Systemd journal, which contains the kernel logs

**journalctl -n n**
Display the most recent n log lines (default is 10)

**journalctl --since "1 hour ago"**
Display events happened in the last hour

**journalctl -x**
Display events, adding explanations from the message catalog

**journalctl -f**
Display the journal in real-time

**journalctl -u crond.service**
Display the log entries created by the cron service

**systemctl restart systemd-journald**
Enable persistent storage of logs in /var/log/journal/ (by default, journalctl stores the logfiles in RAM only)

```
mkdir -p /var/log/journal/ && 
systemctl restart systemd-journald
```
## Kernel compile and patching

### Download

Download the kernel source code `linux-X.Y.Z.tar.bz2` from [http://www.kernel.org](http://www.kernel.org) to the base of the kernel source tree `~/usr/src/linux`

### Clean

- `make clean` Delete most generated files
- `make mrproper` Delete all generated files and kernel configuration
- `make distclean` Delete temporary files, patch leftovers, and similar files

### Configure

- `make config` Create configuration (terminal-based; options must be set in sequence)
- `make menuconfig` Create configuration (ncurses UI)
- `make xconfig` Create configuration (GUI)
- `make gconfig` Create configuration (GUI)
- `make oldconfig` Create a new configuration file, based on the options in the old configuration file and in the source code

Components (e.g. device drivers) can be either:
- not compiled
- compiled into the kernel binary, for support of devices always used on the system or necessary for the system to boot
- compiled as a kernel module, for optional devices

The configuration command creates a configuration file `~/usr/src/linux/.config` containing instructions for the kernel compilation.

### Build

- `make bzImage` Compile the kernel
- `make modules` Compile the kernel modules
- `make all` Compile kernel and kernel modules
- `make -j2 all` will speed up compilation by allocating 2 simultaneous compile jobs

### Modules install

- `make modules_install` Install the previously built modules present in `/lib/modules/X.Y.Z`
- `make install` Install the kernel automatically

### Kernel install

To install the kernel by hand:

1. Copy the new compiled kernel and other files into the boot partition:
   
   ```
   cp ~/usr/src/linux/arch/boot/bzImage /boot/vmlinuz-X.Y.Z (kernel)
   cp ~/usr/src/linux/arch/boot/System.map-X.Y.Z /boot
   cp ~/usr/src/linux/arch/boot/config-X.Y.Z /boot (config options used for this compile)
   ```

2. Create an entry in GRUB to boot on the new kernel:

### Package

Optionally, the kernel can be packaged for install on other machines:

- `make rpm-pkg` Build source and binary RPM packages
- `make binrpm-pkg` Build binary RPM package
- `make deb-pkg` Builds binary DEB package

## Kernel patching

### Download

Download and decompress the patch to `~/usr/src`

### Patch

- `patch -p1 < file.patch` Apply the patch
- `patch -Rp1 < file.patch` Remove (reverse) a patch. Alternatively, applying the patch again reverses it

### Build

Build the patched kernel as explained above

### Install

Install the patched kernel as explained above
Kernel modules allow the kernel to access functions (symbols) for kernel services e.g. hardware drivers, network stack, or filesystem abstraction.

- **lsmod**
  - List the modules that are currently loaded into the kernel

- **insmod module**
  - Insert a module into the kernel. If the module requires another module or if it does not detect compatible hardware, insertion will fail

- **rmmod module**
  - Remove a module from the kernel. If the module is in use by another module, it is necessary to remove the latter module first

- **modinfo module**
  - Display the list of parameters accepted by the module

- **depmod -a**
  - Probe all modules in the kernel modules directory and generate the file that lists their dependencies

- **modprobe module option=value**
  - Insert a module into the running kernel, with the specified parameters. Prerequisite modules will be inserted automatically

- **modprobe -a**
  - Insert all modules

- **modprobe -t directory**
  - Attempt to load all modules contained in the directory until a module succeeds. This action probes the hardware by successive module-insertion attempts for a single type of hardware, e.g. a network adapter

- **modprobe -r module**
  - Remove a module

- **modprobe -c module**
  - Display module configuration

- **modprobe -l**
  - List loaded modules

It is recommended to use **modprobe** instead of **insmod** and **rmmod**, because **modprobe** automatically handles prerequisites when inserting modules, is more specific about errors, and accepts just the module name alone instead of requiring the full pathname.

### Configuration of device drivers

<table>
<thead>
<tr>
<th>Device driver compiled into the kernel</th>
<th>Device driver provided as a kernel module</th>
</tr>
</thead>
</table>
| Configure the device driver by passing a kernel parameter in the GRUB menu: `kernel /vmlinuz ro root=/dev/vg0/root vga=0x33c` | Edit module configuration in `etc/modprobe.conf` or `etc/modprobe.d/` (Red Hat):
  - `alias eth0 3c59x` Specify that eth0 uses the `3c59x.ko` driver module
  - `options 3c509 irq=10,11` Assign IRQ 10 and 11 to 3c509 devices |

---

© Daniele Raffo  dr0.ch
/proc is a pseudo filesystem that gives access to process data held in the kernel.

<table>
<thead>
<tr>
<th>File</th>
<th>Information stored (can be viewed via cat)</th>
<th>Equivalent command</th>
</tr>
</thead>
<tbody>
<tr>
<td>/proc/bus</td>
<td>Buses (e.g. PCI, USB, PC Card)</td>
<td></td>
</tr>
<tr>
<td>/proc/cpupinfo</td>
<td>CPUs information</td>
<td></td>
</tr>
<tr>
<td>/proc/devices</td>
<td>Drivers currently loaded</td>
<td></td>
</tr>
<tr>
<td>/proc/dma</td>
<td>DMA channels in use</td>
<td></td>
</tr>
<tr>
<td>/proc/filesystems</td>
<td>Filesystems supported by the system</td>
<td></td>
</tr>
<tr>
<td>/proc/interrupts</td>
<td>Current IRQs (Interrupt Requests)</td>
<td>procinfo</td>
</tr>
<tr>
<td>/proc/ioports</td>
<td>I/O addresses in use</td>
<td></td>
</tr>
<tr>
<td>/proc/kcore</td>
<td>Memory allocatable by the kernel</td>
<td></td>
</tr>
<tr>
<td>/proc/loadavg</td>
<td>System load averages</td>
<td>uptime</td>
</tr>
<tr>
<td>/proc/mdstat</td>
<td>Information about RAID arrays and devices</td>
<td></td>
</tr>
<tr>
<td>/proc/meminfo</td>
<td>Total and free memory</td>
<td>free</td>
</tr>
<tr>
<td>/proc/modules</td>
<td>Kernel modules currently loaded</td>
<td>lsmod</td>
</tr>
<tr>
<td>/proc/mounts</td>
<td>Mounted partitions</td>
<td>mount</td>
</tr>
<tr>
<td>/proc/net/dev</td>
<td>Network interface statistics</td>
<td></td>
</tr>
<tr>
<td>/proc/partitions</td>
<td>Drive partition information</td>
<td>fdisk -l</td>
</tr>
<tr>
<td>/proc/swaps</td>
<td>Size of total and used swap areas</td>
<td>swapon -s</td>
</tr>
<tr>
<td>/proc/sys/</td>
<td>$sysfs: exposes tunable kernel parameters</td>
<td></td>
</tr>
<tr>
<td>/proc/sys/kernel/</td>
<td>Kernel information and parameters</td>
<td></td>
</tr>
<tr>
<td>/proc/sys/net/</td>
<td>Network information and parameters</td>
<td></td>
</tr>
<tr>
<td>/proc/uptime</td>
<td>Time elapsed since boot</td>
<td>uptime</td>
</tr>
<tr>
<td>/proc/version</td>
<td>Linux version</td>
<td>uname -a</td>
</tr>
<tr>
<td>/proc/n/</td>
<td>Information about process with PID n</td>
<td></td>
</tr>
<tr>
<td>/proc/n/cmdline</td>
<td>Command by which the process was launched</td>
<td>ps n</td>
</tr>
<tr>
<td>/proc/n/cwd</td>
<td>Symlink to process' working directory</td>
<td></td>
</tr>
<tr>
<td>/proc/n/environ</td>
<td>Values of environment variables of process</td>
<td></td>
</tr>
<tr>
<td>/proc/n/exe</td>
<td>Symlink to process' executable</td>
<td></td>
</tr>
<tr>
<td>/proc/n/fd</td>
<td>Files currently opened by the process</td>
<td>lssof -p n</td>
</tr>
<tr>
<td>/proc/n/root</td>
<td>Symlink to process' filesystem root</td>
<td></td>
</tr>
<tr>
<td>/proc/n/status</td>
<td>Status of process</td>
<td></td>
</tr>
</tbody>
</table>

/proc/sys is the only writable branch of /proc and can be used to tune kernel parameters on the fly. All changes are lost after system shutdown, unless applied via `sysctl -p`.

- `sysctl fs.file-max` Get the maximum allowed number of open files
- `cat /proc/sys/fs/file-max` Set the maximum allowed number of open files to 100000
- `sysctl -w "fs.file-max=100000" echo "100000" > /proc/sys/fs/file-max` List all available kernel tuning options
- `sysctl -a` Apply all tuning settings listed in `/etc/sysctl.conf`. This command is usually run at boot by the system initialization script, to make permanent changes to kernel parameters
/dev contains the device files to access all devices in the system.

<table>
<thead>
<tr>
<th>File</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/sda</td>
<td>SCSI, PATA, or SATA hard drive</td>
</tr>
<tr>
<td>/dev/hda</td>
<td>IDE hard drive</td>
</tr>
<tr>
<td>/dev/pda</td>
<td>Parallel port IDE hard drive</td>
</tr>
<tr>
<td>/dev/vda</td>
<td>Virtual disk for KVM-based virtual machines</td>
</tr>
<tr>
<td>/dev/sda, /dev/sdb, /dev/sdc ...</td>
<td>First, second, third ... hard drive</td>
</tr>
<tr>
<td>/dev/sdal, /dev/sda2, /dev/sda3 ...</td>
<td>First, second, third ... partition of the first hard drive</td>
</tr>
<tr>
<td>/dev/md0</td>
<td>Metadisk group, for use with RAID</td>
</tr>
<tr>
<td>/dev/sr0</td>
<td>SCSI CD-ROM</td>
</tr>
<tr>
<td>/dev/pcd0</td>
<td>Parallel port CD-ROM</td>
</tr>
<tr>
<td>/dev/cdrom</td>
<td>CD-ROM. Usually symlinked to /dev/sr0</td>
</tr>
<tr>
<td>/dev/fd0</td>
<td>Floppy disk drive</td>
</tr>
<tr>
<td>/dev/ht0</td>
<td>IDE tape drive</td>
</tr>
<tr>
<td>/dev/pt0</td>
<td>Parallel port tape drive</td>
</tr>
<tr>
<td>/dev/sg0</td>
<td>Generic SCSI device</td>
</tr>
<tr>
<td>/dev/loop0</td>
<td>Loopback device</td>
</tr>
<tr>
<td>/dev/autofs</td>
<td>AutoFS device</td>
</tr>
<tr>
<td>/dev/fuse</td>
<td>FUSE device</td>
</tr>
<tr>
<td>/dev/dsp</td>
<td>Digital Signal Processor device. Interfaces with the soundcard</td>
</tr>
<tr>
<td>/dev/fb0</td>
<td>Framebuffer device. Interfaces with the graphics hardware</td>
</tr>
<tr>
<td>/dev/1p0</td>
<td>Parallel port printer device</td>
</tr>
<tr>
<td>/dev/parport0</td>
<td>Raw parallel port device</td>
</tr>
<tr>
<td>/dev/mem</td>
<td>Physical memory</td>
</tr>
<tr>
<td>/dev/kmem</td>
<td>Kernel virtual memory</td>
</tr>
<tr>
<td>/dev/core</td>
<td>Obsolete. Symlink to /proc/kcore</td>
</tr>
<tr>
<td>/dev/stdin</td>
<td>Standard Input</td>
</tr>
<tr>
<td>/dev/stdout</td>
<td>Standard Output</td>
</tr>
<tr>
<td>/dev/stderr</td>
<td>Standard Error</td>
</tr>
<tr>
<td>/dev/null</td>
<td>Null device, aka blackhole or bit bucket. Discards any received data</td>
</tr>
<tr>
<td>/dev/zero</td>
<td>Zero device. Outputs an infinite stream of zero bytes (NUL) on reads</td>
</tr>
<tr>
<td>/dev/full</td>
<td>&quot;Always full&quot; device. Similar to /dev/zero, and also returns an error &quot;No space left on device&quot; (ENOSPC) on writes</td>
</tr>
<tr>
<td>/dev/random</td>
<td>Non-deterministic random number generator. Gathers entropy from the system to generate randomness; once the entropy pool is depleted, the device blocks all reads until it can collect more entropy</td>
</tr>
<tr>
<td>/dev/urandom</td>
<td>Pseudo random number generator. Faster but unsafe for cryptographic purposes</td>
</tr>
<tr>
<td>/dev/console</td>
<td>System console</td>
</tr>
<tr>
<td>/dev/tty</td>
<td>Terminal for current process</td>
</tr>
<tr>
<td>/dev/tty0</td>
<td>Current virtual console</td>
</tr>
<tr>
<td>/dev/ttyS0</td>
<td>Serial port, usually used for modem connections</td>
</tr>
<tr>
<td>/dev/ptyp0</td>
<td>Pseudo-TTY master</td>
</tr>
<tr>
<td>/dev/ttyp0</td>
<td>Pseudo-TTY slave</td>
</tr>
</tbody>
</table>
If the kernel has been booted in emergency mode and `init` has not been run, some initial configuration is necessary e.g.

```
mount /proc
mount -o remount,rw /
mount -a
```

If mounting the filesystems fails:

```
mknod /dev/sda
mknod /dev/sda1
fdisk -l /dev/sda
fsck -y /dev/sda1
mount -t ext3 /dev/sda1 /mnt/sysimage
chroot /mnt/sysimage
```

To install a package using an alternative root directory (useful if the system has been booted from a removable media):

```
rpm -U --root /mnt/sysimage package.rpm
```

To install GRUB on the specified directory (which must contain `/boot/grub/`):

```
grub-install --root-directory=/mnt/sysimage /dev/sda
```

Alternative method:

```
chroot /mnt/sysimage
grub-install /dev/sda
```

Run `sync` and unmount all filesystems before exiting the shell, to ensure that all changes have been written on disk.

**How to reset the root password (RHEL 7 and 8)**

1. Power up the system and, once on the GRUB 2 boot screen, press `E` to edit the current entry
2. On the kernel line that mentions `linux16`, remove the `rhgb` and `quiet` parameters and add `rd.break` at the end
3. Press `CTRL + X`; the system will boot on the initramfs `switch_root` prompt
4. Remount the filesystem as writable

```
mount -o remount,rw /sysroot
```

5. Change the filesystem root

```
chroot /sysroot
```

6. Modify the root password

```
passwd root
```

7. Force SELinux to relabel context on next boot

```
touch ~/.autorelabel
```

8. Remount the filesystem as readonly (not strictly necessary)

```
mount -o remount,ro /sysroot
```

9. Exit the chroot environment

```
exit
```

10. Resume system boot

```
exit
```
Domain Name System (DNS) is a decentralized hierarchical naming system, mostly used to resolve domain names to IP addresses. It uses TCP and UDP port 53.

<table>
<thead>
<tr>
<th>DNS implementations</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIND</td>
</tr>
<tr>
<td>Unbound</td>
</tr>
<tr>
<td>dnsmasq</td>
</tr>
<tr>
<td>djbdns</td>
</tr>
<tr>
<td>PowerDNS</td>
</tr>
</tbody>
</table>

**named**  BIND Name Daemon

**ndc**    Name Daemon Controller for BIND 8

**rndc**  Remote Name Daemon Controller for BIND 9, uses a shared key to communicate securely with named

```
named -u named -g named  Run BIND as user/group "named" (must be created if needed) instead of root
named -t /var/cache/bind  Run BIND in a chroot jail /var/cache/bind
                         (actually it is the chroot command that starts the named server)
names walk example.org.  DNS debugger
rdcd reconfig            Reload BIND configuration and new zones
rdcd reload example.org  Reload the zone example.org
rdcd freeze example.org  Suspend updates for the zone example.org
rdcd thaw example.org    Resume updates for the zone example.org
rdcd tsig-list           List all currently active TSIG keys
```
DNSSEC was designed to secure the DNS tree and hence prevent cache poisoning. The TSIG (Transaction SIGnature) standard, which authenticates communications between two trusted systems, is used to sign zone transfers and DDNS (Dynamic DNS) updates.

```bash
dnssec-keygen -a dsa -b 1024 \  -n HOST dns1.example.org
```
Generate a TSIG key with DNSSEC algorithm \texttt{nnn} and key fingerprint \texttt{fffff}.
This will create two key files

- `Kdns1.example.org.+nnn+fffff.key`
- `Kdns1.example.org.+nnn+fffff.private`

which contain a key number that must be inserted both in `/etc/named.conf` and `/etc/rndc.conf`

```bash
rndc-confgen -a
```
Generate a `/etc/rndc.key` key file:

```bash
key "rndc-key" {
  algorithm hmac-md5;
  secret "vy2qL3tPHsqnA57e4LT0Ek==";
};
options {
  default-key "rndc-key";
  default-server 127.0.0.1;
  default-port 953;
};
```
This file is automatically read both by `named` and `rndc`

```bash
dnssec-signzone example.org
```
Sign the zone `example.org`
DNS configuration

The configuration file for a DNS server is usually located in `/etc/named.conf`. It contains various options and definitions for the server's behavior and functionality. Here is an example of such a configuration file:

```plaintext
controls {
    inet 127.0.0.1 allow {localhost;}; keys {rndckey;};
};
key "rndc-key" {
    algorithm dsa;
    secret "HYZur46fftdUQ43BJKI093t4t78lkp";
};

acl "mynetwork" {10.7.0.0/24;};

options {
    directory "/var/named";
    version "0.0";
    listen-on port 53 {10.7.0.1; 127.0.0.1;};
    blackhole {172.17.17.0/24;};
    allow-query {mynetwork;};
    allow-query-on {any;};
    allow-query-cache {any;};
    allow-recursion {mynetwork;};
    allow-recursion-on {mynetwork;};
    allow-transfer {10.7.0.254;};
    allow-update {any;};
    recursive-clients 1000;
    dnssec-enable yes;
    dialup no;
    forwarders {10.7.0.252; 10.7.0.253;};
    forward first;
    forwarders {10.7.0.254;};
};

// Define the root name servers
zone "." {
    type hint;
    file "root.cache";
}

// Configure system to act as a master server for the example.org domain
zone "example.org" IN {
    type master;
    file "master/example.org.zone";
};
zone "240.123.224.in-addr.arpa" IN {
    type master;
    file "slave/example.org.revzone";
};

// Configure system to act as a slave server for the example2.org domain
zone "example2.org" IN {
    type slave;
    masters {10.7.0.254;};
};
zone "0.7.10.in-addr.arpa" IN {
    type slave;
    file "slave/10.7.0.revzone";
};
```

This configuration file includes various options and zones, which control the server's functionality. For instance, the `controls` section specifies which hosts are allowed to access the DNS server and which keys are used for securing communication. The `options` section contains server-wide settings such as the directory for the working files, version information, listen ports, blacklisted IP addresses, and permissions to allow or deny queries and updates. The `zones` section defines individual DNS zones, which map domain names to IP addresses.
### DNS zone file

```
$TTL 86400                      ; TTL (1 day)
$ORIGIN example.org.
example.org IN SOA dns1.example.org. help.example.org. (   ; Master DNS server is dns1.example.org
  2014052300 ; serial           ; If problems, contact help@example.org
  28800 ; refresh (8 hours)
  7200 ; retry (2 hours)
  604800 ; expire (1 week)
  600 ) ; negative TTL (10 mins)

IN NS dns1.example.org.
IN NS dns2.example.org.
IN MX 10 mail1.example.org.
IN MX 20 mail2.example.org.
dns1 IN A 224.123.240.3
dns2 IN A 224.123.240.4
mail1 IN A 224.123.240.73
mail2 IN A 224.123.240.77
foo IN A 224.123.240.12
bar IN A 224.123.240.13
www IN A 224.123.240.19
baz IN CNAME bar

subdomain IN NS ns1.subdomain.example.org. ; Glue records
IN NS ns2.subdomain.example.org.
ns1.subdomain.example.org. IN A 224.123.240.201
ns2.subdomain.example.org. IN A 224.123.240.202
```

### DNS reverse zone file

```
$TTL 86400                      ; TTL (1 day)
example.org IN SOA dns1.example.org. help.example.org. (   ; Master DNS server is dns1.example.org
  2014052300 ; serial           ; If problems, contact help@example.org
  28800 ; refresh (8 hours)
  7200 ; retry (2 hours)
  604800 ; expire (1 week)
  600 ) ; negative TTL (10 mins)

12.240.123.224.in-addr.arpa IN PTR foo
13.240.123.224.in-addr.arpa IN PTR bar
19.240.123.224.in-addr.arpa IN PTR www
```

### Resource Records

- **$TTL**: How long to cache a positive response
- **$ORIGIN**: Suffix appended to all names not ending with a dot. Useful when defining multiple subdomains inside the same zone
- **SOA**: Start Of Authority for the example.org zone
  - **serial**: Serial number. Must be increased after each edit of the zone file
  - **refresh**: How frequently a slave server refreshes its copy of zone data from the master
  - **retry**: How frequently a slave server retries connecting to the master
  - **expire**: How long a slave server relies on its copy of zone data. After this time period expires, the slave server is not authoritative anymore for the zone unless it can contact a master
  - **negative TTL**: How long to cache a non-existent answer
- **A**: Address: maps names to IP addresses. Used for DNS lookups.
- **PTR**: Pointer: maps IP addresses to names. Used for reverse DNS lookups. Each A record must have a matching PTR record
- **CNAME**: Canonical Name: specifies an alias for a host with an A record (even in a different zone). Discouraged as it causes multiple lookups; it is better to use multiple A records instead
- **NS**: Name Service: specifies the authoritative name servers for the zone
- **MX**: Mailserver: specifies address and priority of the servers able to handle mail for the zone

Glue Records are not really part of the zone; they delegate authority for other zones, usually subdomains.
## HTTP response codes

| 1XX Informational | 100 Continue | The server received the request headers, so the client should continue by sending the remainder of the request |
| 101 Switching Protocols | The server agreed to switch protocol upon client's demand |
| 2XX Success | 200 OK | The request was successful |
| 201 Created | The request was successful, and resulted in a resource being created |
| 204 No Content | The request was successful, and the server does not need to return any content |
| 206 Partial Content | The request was successful, and the server is returning only partial content because the client sent a Range header field |
| 3XX Redirection | 301 Moved Permanently | The requested resource was permanently moved to a new URI |
| 302 Found | The requested resource was temporarily moved to a new URI |
| 303 See Other | The requested resource can be found on another URI, and should be retrieved from there via a GET |
| 304 Not Modified | The client sent a conditional GET request, and the resource has not been modified since last time it was requested |
| 307 Temporary Redirect | The requested resource was temporarily moved to a new URI, but future requests should use the original URI |
| 4XX Client Error | 400 Bad Request | The server was unable to understand the request due to bad syntax |
| 401 Unauthorized | The request requires user authentication |
| 403 Forbidden | The client did not have the necessary permissions to access the requested resource |
| 404 Not Found | The requested resource was not found on the server |
| 408 Request Timeout | The server timed out while waiting for the request |
| 409 Conflict | The request could not be processed because of a conflict in the resource state |
| 410 Gone | The requested resource is no longer available on the server and will not be available again |
| 451Unavailable for Legal Reasons | The requested resource is not available due to government censorship |
| 5XX Server Error | 500 Internal Server Error | The server encountered a generic error while trying to fulfill the request |
| 501 Not Implemented | The server was unable to recognize the request method |
| 502 Bad Gateway | The server is acting as a gateway or proxy, and received an invalid response from the upstream server |
| 503 Service Unavailable | The server is temporarily unavailable due to overload or maintenance |
| 504 Gateway Timeout | The server is acting as a gateway or proxy, and a request to the upstream server timed out |
| 505 HTTP Version Not Supported | The server does not support the HTTP protocol version used in the request |
Apache is an open source and widespread HTTP server, originally based on the NCSA HTTPd server.

/etc/httpd/conf/httpd.conf (Red Hat) Apache configuration files
/etc/httpd/conf.d/*.conf (Debian and SUSE)
/etc/apache2/httpd.conf

/var/www/html Default document root directory
$HOME/public_html Default document root directory for users’ websites

Web content must be readable by the user/group the Apache process runs as. For security reasons, it should be owned and writable by the superuser or the webmaster user/group (usually www-data), not the Apache user/group.

apachectl (Red Hat) Manage the Apache webserver
httpd (Red Hat)
apache2ctl (Debian)
apachectl start Start the Apache webserver daemon
apachectl status Display a brief status report
apachectl fullstatus Display a detailed status report
apachectl graceful Gracefully restart Apache; currently open connections are not aborted
apachectl graceful-stop Gracefully stop Apache; currently open connections are not aborted
apachectl configtest Test the configuration file, reporting any syntax error
apachectl -t
apachectl -M List all loaded and shared modules

The Apache webserver contains a number of MPMs (Multi-Processing Modules) which can operate following two methods:

<table>
<thead>
<tr>
<th>MPM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefork MPM</td>
<td>A number of child processes is spawned in advance, with each child serving one connection. Highly reliable due to Linux memory protection that isolates each child process.</td>
</tr>
<tr>
<td>worker MPM</td>
<td>Multiple child processes spawn multiple threads, with each thread serving one connection. More scalable but prone to deadlocks if third-party non-threadsafe modules are loaded.</td>
</tr>
</tbody>
</table>

### HTTPS

HTTPS (i.e. HTTP over SSL/TLS) allows securing communications between the webserver and the client by encrypting all communications end-to-end between the two. A webserver using HTTPS hands over its public key to the client when the client connects to the server via port 443. The server’s public key is signed by a CA (Certification Authority), whose validity is ensured by the root certificates stored into the client’s browser.

The `openssl` command and its user-friendly `CA.pl` script are the tools of the OpenSSL cryptographic library that can be used to accomplish all public key crypto operations e.g. generate key pairs, Certificate Signing Requests, and self-signed certificates. Another user-friendly tool is `genkey`.

Virtual hosting with HTTPS requires assigning a unique IP address for each virtual host; this because the SSL handshake (during which the server sends its certificate to the client’s browser) takes place before the client sends the `Host` header (which tells to which virtual host the client wants to talk).

A workaround for this is SNI (Server Name Indication) which makes the browser send the hostname in the first message of the SSL handshake. Another workaround is to have all multiple name-based virtual hosts use the same SSL certificate with a wildcard domain e.g. *.example.org.
<table>
<thead>
<tr>
<th><strong>Apache configuration file</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server configuration directives</strong></td>
</tr>
<tr>
<td><strong>ServerName <a href="http://www.mysite.org:80">www.mysite.org:80</a></strong></td>
</tr>
<tr>
<td><strong>ServerRoot /etc/httpd</strong></td>
</tr>
<tr>
<td><strong>ServerAdmin <a href="mailto:webmaster@mysite.org">webmaster@mysite.org</a></strong></td>
</tr>
<tr>
<td><strong>StartServers 5</strong></td>
</tr>
<tr>
<td><strong>MinSpareServers 5</strong> <strong>MaxSpareServers 10</strong></td>
</tr>
<tr>
<td><strong>MaxClients 256</strong> <strong>MaxRequestWorkers 256</strong> (<strong>v2.3.13 and later</strong>)<strong>transported</strong></td>
</tr>
<tr>
<td><strong>ServerLimit 256</strong></td>
</tr>
<tr>
<td><strong>ThreadsPerChild 25</strong></td>
</tr>
<tr>
<td><strong>ThreadLimit 64</strong></td>
</tr>
<tr>
<td><strong>MaxRequestsPerChild 16</strong> (<strong>v2.2</strong>)<strong>transported</strong></td>
</tr>
<tr>
<td><strong>MaxConnectionsPerChild 16</strong> (<strong>v2.4</strong>)<strong>transported</strong></td>
</tr>
<tr>
<td><strong>LoadModule mime_module modules/mod_mime.so</strong></td>
</tr>
<tr>
<td><strong>Listen 10.17.1.1:80</strong> <strong>Listen 10.17.1.5:8080</strong></td>
</tr>
<tr>
<td><strong>User nobody</strong> <strong>Group nobody</strong></td>
</tr>
</tbody>
</table>
### Apache configuration file

**Main configuration directives**

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocumentRoot</td>
<td>/var/www/html</td>
</tr>
<tr>
<td>Alias</td>
<td>/image /mydir/pub/image</td>
</tr>
<tr>
<td>TypesConfig</td>
<td>conf/mime.types</td>
</tr>
<tr>
<td>AddType</td>
<td>image/jpeg jpg jpe</td>
</tr>
<tr>
<td>Redirect</td>
<td>/foo /bar</td>
</tr>
<tr>
<td>Redirect</td>
<td>/foo <a href="http://www.example.com/foo">http://www.example.com/foo</a></td>
</tr>
<tr>
<td>AccessFileName</td>
<td>.htaccess</td>
</tr>
<tr>
<td>&lt;Directory&gt;</td>
<td>/var/www/html/foobar</td>
</tr>
<tr>
<td>AllowOverride</td>
<td>AuthConfig Limit</td>
</tr>
<tr>
<td>Logging directives</td>
<td></td>
</tr>
<tr>
<td>LogFormat</td>
<td>&quot;%h %l %u %t &quot;%r&quot; %&gt;s %b&quot; common</td>
</tr>
<tr>
<td>CustomLog</td>
<td>/var/log/httpd/access_log common</td>
</tr>
<tr>
<td>TransferLog</td>
<td>/var/log/httpd/access_log</td>
</tr>
<tr>
<td>TransferLog</td>
<td>&quot;</td>
</tr>
<tr>
<td>HostnameLookups</td>
<td>Off</td>
</tr>
</tbody>
</table>

**Limited scope directives**

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Directory&gt;</td>
<td>/var/www/html/foobar</td>
</tr>
<tr>
<td>AllowOverride</td>
<td>AuthConfig Limit</td>
</tr>
<tr>
<td>Logging directives</td>
<td></td>
</tr>
<tr>
<td>LogFormat</td>
<td>&quot;%h %l %u %t &quot;%r&quot; %&gt;s %b&quot; common</td>
</tr>
<tr>
<td>CustomLog</td>
<td>/var/log/httpd/access_log common</td>
</tr>
<tr>
<td>TransferLog</td>
<td>/var/log/httpd/access_log</td>
</tr>
<tr>
<td>TransferLog</td>
<td>&quot;</td>
</tr>
<tr>
<td>HostnameLookups</td>
<td>Off</td>
</tr>
</tbody>
</table>

**LogFormat**

- In this case, specifies "common" for the CLF (Common Log Format) which is defined as such:
  - %h: IP address of the client host
  - %l: Identity of client as determined by identd
  - %u: User ID of client making the request
  - %t: Timestamp the server completed the request
  - %r: Request as done by the user
  - %s: Status code sent by the server to the client
  - %b: Size of the object returned, in bytes

**Logging directives**

- CustomLog: Set up a log filename, with the format or (as in this case) the nickname specified
- TransferLog: Set up a log filename, with format determined by the most recent LogFormat directive which did not define a nickname
- TransferLog "|rotatelogs access_log 86400": Set log rotation every 24 hours
- HostnameLookups: Disable DNS hostname lookup to save network traffic. Hostnames can be resolved later by processing the log file: logresolve <access_log>accessdns_log
### Apache configuration file

<table>
<thead>
<tr>
<th>Virtual hosts directives</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameVirtualHost * (v2.2)</td>
</tr>
<tr>
<td>Specify which IP address will serve virtual hosting. The argument can be an IP address, an address:port pair, or * for all IP addresses of the server. The same argument need to be inserted in the relevant &lt;VirtualHost&gt; directive</td>
</tr>
</tbody>
</table>

```
<VirtualHost *:80>
  ServerName www.mysite.org
  ServerAlias mysite.org *.mysite.org
  DocumentRoot /var/www/vhosts/mysite
</VirtualHost>
```

The first listed virtual host is also the default virtual host. It inherits those main settings that does not override. This virtual host answers to http://www.mysite.org, and also redirects there all HTTP requests on the domain mysite.org.

```
<VirtualHost *:80>
  ServerAdmin webmaster@www.mysite2.org
  ServerName www.mysite2.org
  DocumentRoot /var/www/vhosts/mysite2
  ErrorLog /var/www/logs/mysite2
</VirtualHost>
```

Name-based virtual host http://www.mysite2.org. Multiple name-based virtual hosts can share the same IP address; DNS must be configured accordingly to map each name to the correct IP address. Cannot be used with HTTPS.

```
<VirtualHost *:8080>
  ServerName www.mysite3.org
  DocumentRoot /var/www/vhosts/mysite3
</VirtualHost>
```

Port-based virtual host answering to connections on port 8080. A Listen 8080 directive must also be present.

```
<VirtualHost 10.17.1.5:80>
  ServerName www.mysite4.org
  DocumentRoot /var/www/vhosts/mysite4
</VirtualHost>
```

IP-based virtual host answering to http://10.17.1.5.
### Apache configuration file

<table>
<thead>
<tr>
<th>Authorization directives</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthName &quot;Protected zone&quot;</td>
<td>Name of the realm. The client will be shown the realm name and prompted to enter a user and password</td>
</tr>
<tr>
<td>AuthType Basic</td>
<td>Type of user authentication: Basic, Digest, Form, or None</td>
</tr>
<tr>
<td>AuthUserFile &quot;/var/www/.htpasswd&quot;</td>
<td>User database file. Each line has the format user:encryptedpassword. To add a user to the database file, use the command: htpasswd /var/www/.htpasswd user (will prompt for a password)</td>
</tr>
<tr>
<td>AuthGroupFile &quot;/var/www/.htgroup&quot;</td>
<td>Group database file. Each line specifies a group followed by the usernames of all its members: group: user1 user2 user3</td>
</tr>
<tr>
<td>Require valid-user</td>
<td>Control who can access the protected resource. valid-user Any user in the user database file user user Only the specified user group group Only the members of the specified group</td>
</tr>
<tr>
<td>Satisfy Any</td>
<td>Set the access policy concerning user and host control. All Both Require and Allow criteria must be satisfied Any Any of Require or Allow criteria must be satisfied</td>
</tr>
<tr>
<td>Allow from 10.13.13.0/24 Deny from 10.13.14.0/24 (deprecated)</td>
<td>Control which host can or cannot access the protected resource</td>
</tr>
<tr>
<td>Order Allow,Deny (deprecated)</td>
<td>Control the evaluation order of Allow and Deny directives. Allow,Deny First, all Allow directives are evaluated; at least one must match, or the request is rejected. Next, all Deny directives are evaluated; if any matches, the request is rejected. Last, any requests which do not match an Allow or a Deny directive are denied Deny,Allow First, all Deny directives are evaluated; if any match, the request is denied unless it also matches an Allow directive. Any requests which do not match any Allow or Deny directives are permitted</td>
</tr>
</tbody>
</table>
### Apache configuration file

#### SSL/TLS directives (mod_ssl module)

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSLCertificateFile \</td>
<td>SSL server certificate</td>
</tr>
<tr>
<td>/etc/httpd/conf/ssl.crt/server.crt</td>
<td></td>
</tr>
<tr>
<td>SSLCertificateKeyFile \</td>
<td>SSL server private key (for security reasons, this file must be</td>
</tr>
<tr>
<td>/etc/httpd/conf/ssl.key/server.key</td>
<td>mode 600 and owned by root)</td>
</tr>
<tr>
<td>SSLCACertificatePath \</td>
<td>Directory containing the certificates of CAs. Files in this directory are</td>
</tr>
<tr>
<td>/usr/local/apache2/conf/ssl.crt/</td>
<td>PEM-encoded and accessed via symlinks to hash filenames</td>
</tr>
<tr>
<td>SSLCACertificateFile \</td>
<td>Certificates of CAs. Certificates are PEM-encoded and concatenated in a</td>
</tr>
<tr>
<td>/usr/local/apache2/conf/ssl.crt/ca-bundle.crt</td>
<td>single bundle file in order of preference</td>
</tr>
<tr>
<td>SSLCertificateChainFile \</td>
<td>Certificate chain of the CAs. Certificates are PEM-encoded and concatenated</td>
</tr>
<tr>
<td>/usr/local/apache2/conf/ssl.crt/ca.crt</td>
<td>from the issuing CA certificate of the server certificate to the root CA</td>
</tr>
<tr>
<td></td>
<td>certificate. Optional</td>
</tr>
<tr>
<td>SSLEngine on</td>
<td>Enable the SSL/TLS Protocol Engine</td>
</tr>
<tr>
<td>SSLProtocol +SSLv3 +TLSv1.2</td>
<td>SSL protocol flavors that the client can use to connect to server. Possible</td>
</tr>
<tr>
<td></td>
<td>values are:</td>
</tr>
<tr>
<td></td>
<td>SSLv2 (deprecated)</td>
</tr>
<tr>
<td></td>
<td>SSLv3</td>
</tr>
<tr>
<td></td>
<td>TLSv1</td>
</tr>
<tr>
<td></td>
<td>TLSv1.1</td>
</tr>
<tr>
<td></td>
<td>TLSv1.2</td>
</tr>
<tr>
<td></td>
<td>All (all the above protocols)</td>
</tr>
<tr>
<td>SSLCipherSuite \</td>
<td>Cipher suite available for the SSL handshake (key exchange algorithms,</td>
</tr>
<tr>
<td>ALL:!aDH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP</td>
<td>authentication algorithms, cipher/encryption algorithms, MAC digest</td>
</tr>
<tr>
<td></td>
<td>algorithms)</td>
</tr>
<tr>
<td>ServerTokens Full</td>
<td>Server response header field to send back to client. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>Prod sends Server: Apache</td>
</tr>
<tr>
<td></td>
<td>Major sends Server: Apache/2</td>
</tr>
<tr>
<td></td>
<td>Minor sends Server: Apache/2.4</td>
</tr>
<tr>
<td></td>
<td>Minimal sends Server: Apache/2.4.2</td>
</tr>
<tr>
<td></td>
<td>OS sends Server: Apache/2.4.2 (Unix)</td>
</tr>
<tr>
<td></td>
<td>Full sends Server: Apache/2.4.2 (Unix) \</td>
</tr>
<tr>
<td></td>
<td>PHP/4.2.2 MyMod/1.2 (default)</td>
</tr>
<tr>
<td>ServerSignature Off</td>
<td>Trailing footer line on server-generated documents. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>Off no footer line (default)</td>
</tr>
<tr>
<td></td>
<td>On server version number and ServerName</td>
</tr>
<tr>
<td></td>
<td>EMall as above, plus a mailto link to ServerAdmin</td>
</tr>
<tr>
<td>SSLVerifyClient none</td>
<td>Certificate verification level for client authentication. Possible values</td>
</tr>
<tr>
<td></td>
<td>are:</td>
</tr>
<tr>
<td></td>
<td>none no client certificate is required</td>
</tr>
<tr>
<td></td>
<td>require the client needs to present a valid certificate</td>
</tr>
<tr>
<td></td>
<td>optional the client may present a valid certificate (this option is unused</td>
</tr>
<tr>
<td></td>
<td>it doesn't work on all browsers)</td>
</tr>
<tr>
<td></td>
<td>optional_no_ca the client may present a valid certificate but it doesn't</td>
</tr>
<tr>
<td></td>
<td>need to be successfully verifiable (this option is practically used only</td>
</tr>
<tr>
<td></td>
<td>for SSL testing)</td>
</tr>
<tr>
<td>TraceEnable on</td>
<td>Enable TRACE requests</td>
</tr>
</tbody>
</table>
A **forward proxy** provides proxy services, typically web content caching and/or filtering, for clients located in a LAN. All outgoing requests from the clients, and the responses from the Internet, pass through the proxy. The clients must be manually configured (e.g. in the browser’s connection settings) to use the proxy.

### Apache configuration file

**Forward proxy**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxyRequests On</td>
<td>Enable forward proxy requests</td>
</tr>
<tr>
<td>ProxyVia On</td>
<td>Add a <code>Via: HTTP</code> header line to every request and reply</td>
</tr>
<tr>
<td>&lt;Proxy *</td>
<td>Serve only proxy requests coming from 10.1.1.0/24</td>
</tr>
</tbody>
</table>

**Reverse proxy**

A **reverse proxy** aka **gateway** allows to expose a single entry point for one or more web servers in a LAN. This improves security and simplifies management, as features (e.g. load balancing, firewalling, automatic redirection from HTTP to HTTPS, redirection on default ports) can be configured centrally. It is necessary to create a DNS A record that maps site.example.com to the public IP address of the proxy.

### Apache configuration file

**Reverse proxy**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerName site.example.com</td>
<td>Virtual host for HTTP</td>
</tr>
<tr>
<td>RewriteEngine On</td>
<td>Define website name</td>
</tr>
<tr>
<td>RewriteCond %{HTTPS} off</td>
<td>Redirect all HTTP requests to HTTPS</td>
</tr>
<tr>
<td>RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}</td>
<td>Alternatively: Redirect “/” “<a href="https://10.2.2.73:443/%E2%80%9D">https://10.2.2.73:443/”</a></td>
</tr>
<tr>
<td>&lt;VirtualHost *:443</td>
<td>Virtual host for HTTPS</td>
</tr>
<tr>
<td>ServerName site.example.com</td>
<td>Define website name</td>
</tr>
<tr>
<td>ServerSignature On</td>
<td>Set a footer line under server-generated pages</td>
</tr>
<tr>
<td>&lt;Proxy *&gt;</td>
<td>Serve all proxy requests</td>
</tr>
<tr>
<td>SSLEngine on</td>
<td>Enable and configure SSL</td>
</tr>
<tr>
<td>SSLProtocol ALL -SSLv2 -SSLv3</td>
<td></td>
</tr>
<tr>
<td>SSLHonorCipherOrder on</td>
<td></td>
</tr>
<tr>
<td>SSLCipherSuites DEFAULT</td>
<td></td>
</tr>
<tr>
<td>SSLCertificateFile /etc/ssl/site.crt</td>
<td></td>
</tr>
<tr>
<td>SSLCertificateKeyFile /etc/ssl/site.key</td>
<td></td>
</tr>
<tr>
<td>SSLCACertificateFile /etc/ssl/site.ca.crt</td>
<td></td>
</tr>
<tr>
<td>ProxyPass &quot;/&quot; &quot;<a href="http://10.2.2.73:8080/">http://10.2.2.73:8080/</a>&quot;</td>
<td>Enable reverse proxying for server 10.2.2.73</td>
</tr>
<tr>
<td>ProxyPassReverse &quot;/&quot; &quot;<a href="http://10.2.2.73:8080/">http://10.2.2.73:8080/</a>&quot;</td>
<td></td>
</tr>
<tr>
<td>&lt;/VirtualHost&gt;</td>
<td></td>
</tr>
</tbody>
</table>
Apache Tomcat is an open source Java Servlet Container implementing several Java EE specifications, originally part of the Jakarta Project. It is composed of:
- Catalina, the core component and servlet container implementation;
- Coyote, an HTTP connector component, providing a pure Java webserver environment to run Java code;
- Jasper, a JSP (Java Server Pages) engine, which parses JSP files and compiles them into Java servlets.

Tomcat has been removed from RHEL 8; instead, it is suggested to use the JBoss Enterprise Application Platform, which includes Apache and Tomcat.

$JAVA_HOME Root of the Java installation e.g. /usr/lib/jvm/java-1.8.0-openjdk.x86_64/
$CATALINA_HOME Root of the Tomcat installation e.g. /usr/share/tomcat7/
$CATALINA_BASE Tomcat may also be configured for multiple instances by defining the variable $CATALINA_BASE for each instance. If a single instance of Tomcat is running, $CATALINA_BASE is the same as $CATALINA_HOME

### Tomcat global files

- `$CATALINA_BASE/conf/server.xml`: Tomcat main configuration file
- `$CATALINA_BASE/conf/web.xml`: Options and values applied to all web applications running on a specific Tomcat instance. These can be overridden by the application-specific servlet configuration defined in `$CATALINA_BASE/webapps/appname/WEB-INF/web.xml`
- `$CATALINA_BASE/conf/context.xml`: Context applied to all web applications running on a specific Tomcat instance
- `$CATALINA_BASE/conf/tomcat-users.xml`: Users, passwords, and roles applied to a specific Tomcat instance
- `$CATALINA_BASE/conf/catalina.policy`: Tomcat's core security policy for the Catalina class
- `$CATALINA_BASE/conf/catalina.properties`: Java properties file for the Catalina class
- `$CATALINA_BASE/conf/logging.properties`: Java properties file for Catalina's built-in logging functions
- `$CATALINA_BASE/lib/`: JAR files accessible by both web applications and internal Tomcat code
- `$JAVA_HOME/jre/lib/security/keystore.jks`: Java keystore

### Tomcat application-specific files

- `$CATALINA_BASE/webapps/appname/WEB-INF/`: HTML, JSP, and other files to serve to the client browser
- `$CATALINA_BASE/webapps/appname/WEB-INF/web.xml`: Description of servlets and other components of the application, and initialization parameters
- `$CATALINA_BASE/webapps/appname/WEB-INF/classes/`: Java class files that aren't in JAR format. The directory hierarchy from here reflects the class hierarchy
- `$CATALINA_BASE/webapps/appname/WEB-INF/lib/`: Other JAR files (e.g. third-party libraries, JDBC drivers) required by the application

### Tomcat log files

- `$CATALINA_BASE/logs/catalina.out`: Tomcat log
- `$CATALINA_BASE/logs/localhost.log`: Host log
- `$CATALINA_BASE/logs/localhost_access.log`: Host HTTP access log
- `$CATALINA_BASE/logs/manager.log`: Application log
- `$CATALINA_BASE/logs/host-manager.log`: Application log

`java -X` Display all available `-X` options (nonstandard HotSpot JVM options)
`java -XshowSettings:properties -version` Print Java runtime settings
Samba is a free-software, cross-platform implementation of SMB/CIFS. SMB (Server Message Block) is a Microsoft proprietary protocol for file and printer sharing, while CIFS (Common Internet File System) is the public version of SMB.

### Commonly used ports in Samba

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>137</td>
<td>TCP/UDP</td>
<td>netbios-ns</td>
<td>NetBIOS Name Service requests and responses</td>
</tr>
<tr>
<td>138</td>
<td>TCP/UDP</td>
<td>netbios-dgm</td>
<td>NetBIOS Datagram Service e.g. server announcements</td>
</tr>
<tr>
<td>139</td>
<td>TCP/UDP</td>
<td>netbios-ssn</td>
<td>NetBIOS Session Service e.g. file and printer sharing</td>
</tr>
<tr>
<td>445</td>
<td>TCP</td>
<td>microsoft-ds</td>
<td>Active Directory; registration and translation of NetBIOS names, network browsing</td>
</tr>
<tr>
<td>389</td>
<td>TCP</td>
<td>LDAP</td>
<td></td>
</tr>
<tr>
<td>901</td>
<td>TCP</td>
<td>SWAT service</td>
<td></td>
</tr>
</tbody>
</table>

The full list of used ports can be found via the command `grep -i netbios /etc/services`.

- **smbd**: Server Message Block daemon. Provides SMB file and printer sharing, browser services, user authentication, and resource lock. An extra copy of this daemon runs for each client connected to the server.
- **nmbd**: NetBIOS Name Service daemon. Handles NetBIOS name lookups, WINS requests, list browsing and elections. An extra copy of this daemon runs if Samba functions as a WINS server; another extra copy of this daemon runs if DNS is used to translate NetBIOS names.

- **WINS** (Windows Internet Name Service) is a name service used to translate NetBIOS names to IP addresses.

- **/etc/smb/** (RHEL 7): Samba directory
- **/etc/samba/1mhosts**: Samba NetBIOS hosts file
- **/etc/samba/netlogon**: User logon directory

- **smbd -V**: Show the version of the Samba server
- **smbclient -V**: Show the version of the Samba client

- **testparm**: Check the Samba configuration file and report any error

- **smbpasswd user**: Change the Samba password of `user`
- **smbpasswd -a user**: Create a new Samba `user` and set their password

- **nmblookup smbserver**: Look up the NetBIOS name of a server and map it to an IP address
- **nmblookup -U winsserver -R WORKGROUP#1B**: Query recursively a WINS server for the Domain Master Browser for the specified workgroup
- **nmblookup -U winsserver -R WORKGROUP#1D**: Query recursively a WINS server for the Domain Controller for the specified workgroup

- **net rpc shutdown -r -S smbserver -U root%password**: Reboot a CIFS server
- **net rpc service list -S smbserver**: List available services on a CIFS server
- **net status sessions**: Show active Samba sessions
- **net status shares**: Show Samba shares
- **net rpc info**: Show information about the domain
- **net groupmap list**: Show group mappings between Samba and MS Windows
mount.cifs
smbmount

Mount a Samba share on a Linux filesystem, using the CIFS filesystem interface

```
mount //smbserver/share1 /mnt/share1 \
-t cifs -o username=user
```

Mount a Samba share as `user`

smbstatus

Display current information about shares, clients connections, and locked files

```
smbclient //smbserver/share1
```

Access a Samba share on a server (with an FTP-like interface)

```
smbclient -L //smbserver -W WORKGROUP -U user
```
List the Samba resources available on a server, belonging to the specified workgroup and accessible to the specified user

```
cat msg.txt | smbclient -M client -U user
```
Show a message popup on the client machine, using the WinPopup protocol

---

### Samba mount options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username=USER</td>
<td>Mount the share as <code>USER</code></td>
</tr>
<tr>
<td>password=PASSWORD</td>
<td>Specify the mount user's <code>PASSWORD</code></td>
</tr>
<tr>
<td>credentials=CREDFILE</td>
<td>Mount the share as the user defined in the credentials file <code>CREDFILE</code> which must have this format: username=USER password=PASSWORD</td>
</tr>
<tr>
<td>multiuser</td>
<td>Mount the share in multiuser mode</td>
</tr>
<tr>
<td>sec=ntlmssp</td>
<td>Set the security level to NTLMSSP, This is required in RHEL 7 to enable multiuser mode</td>
</tr>
</tbody>
</table>

© Daniele Raffo

Linux Guide 9th ed., Jan 2021
### Samba - global configuration

```
[global]

workgroup = MYWORKGROUP
server string = Linux Samba Server %L
hosts allow = 10.9.9.0/255.255.255.0
security = user
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd

unix password sync = yes
username map = /etc/samba/smbusers

netbios name = Mysambabox
netbios aliases = Mysambabox1
wins support = yes
logon server = yes
log file = /var/log/samba/log.%m
max log size = 1000
syslog only = no
syslog = 0

panic action = \
/usr/share/samba/panic-action %d
```

- **workgroup = MYWORKGROUP**
  - Make Samba join the specified workgroup

- **server string = Linux Samba Server %L**
  - Describe server to the clients

- **hosts allow = 10.9.9.0/255.255.255.0**
  - Allow only the specified machines to connect to the server

- **security = user**
  - Set up user-level authentication

- **encrypt passwords = yes**
  - Use encrypted passwords

- **smb passwd file = /etc/samba/smbpasswd**
  - Refer to the specified password file for user authentication.
    - A new user’s password will need to be set both in Linux and Samba by running these commands from a shell prompt:
      ```
passwd newuser
smbpasswd newuser
```

- **unix password sync = yes**
  - When the password of a client user (e.g. under MS Windows) is changed, change the Linux and Samba passwords accordingly

- **username map = /etc/samba/smbusers**
  - Map each Samba server user name to client user name(s).
    - The file `/etc/samba/smbusers` has the following format:
      ```
      root = Administrator Admin
      jdoe = "John Doe"
      kgreen = "Kim Green"
      ```

- **netbios name = Mysambabox**
  - Set NetBIOS name and alias

- **netbios aliases = Mysambabox1**

- **wins support = yes**

- **logon server = yes**

- **log file = /var/log/samba/log.%m**
  - Use a separate logfile for each machine that connects

- **max log size = 1000**
  - Maximum size of each logfile, in Kb

- **syslog only = no**

- **syslog = 0**
  - Log everything to the logfiles `/var/log/smb/log.smbd` and `/var/log/smb/log.nmbd`, and log a minimum amount of information to syslog. This parameter can be set to a higher value to have syslog log more information

- **panic action = \n/usr/share/samba/panic-action %d**
  - Mail a backtrace to the sysadmin in case Samba crashes

### [netlogon]

```
[netlogon]

comment = Netlogon for Windows clients
path = /home/netlogon
logon script = %U.bat

browseable = no
writeable = no
guest ok = no
```

- **comment = Netlogon for Windows clients**

- **path = /home/netlogon**

- **logon script = %U.bat**
  - Specifies a per-user script e.g. `/home/netlogon/jdoe.bat` will be called when user jdoe logs in.

- **browseable = no**

- **writeable = no**

- **guest ok = no**

### [Canon LaserJet 3]

```
[Canon LaserJet 3]

printer name = lp
comment = Canon LaserJet 3 main printer
path = /var/spool/lpd/samba
printable = yes
writeable = no
```

- **printer name = lp**
  - Specifies a per-clientname script e.g. `/var/spool/lpd/samba` will be called when a specific machine logs in.

- **comment = Canon LaserJet 3 main printer**

- **path = /var/spool/lpd/samba**

- **printable = yes**

- **writeable = no**
### Samba - share configuration

<table>
<thead>
<tr>
<th>Section</th>
<th>smb.conf</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[public]</td>
<td></td>
<td>Section defining a public share accessible on read/write by anyone</td>
</tr>
<tr>
<td></td>
<td>comment = Public Storage on %L</td>
<td>Describe the public share to users</td>
</tr>
<tr>
<td></td>
<td>path = /home/samba</td>
<td>Path of the public share on the server</td>
</tr>
<tr>
<td></td>
<td>browsable = yes</td>
<td>Show the public share when browsing</td>
</tr>
<tr>
<td></td>
<td>writeable = yes</td>
<td>Allow all users to write in this directory</td>
</tr>
<tr>
<td>[homes]</td>
<td></td>
<td>Section enabling users that have an account and a home directory on the Samba server to access it and modify its contents from a Samba client. The path variable is not set, by default is path=/home/%S</td>
</tr>
<tr>
<td></td>
<td>comment = %U's home directory on %L from %m</td>
<td>Describe the share to the user</td>
</tr>
<tr>
<td></td>
<td>browsable = no</td>
<td>Do not show the homes share when browsing</td>
</tr>
<tr>
<td></td>
<td>writeable = yes</td>
<td>Allow the user to write in their home directory</td>
</tr>
<tr>
<td>[foobar]</td>
<td></td>
<td>Section defining a specific share</td>
</tr>
<tr>
<td></td>
<td>path = /foobar</td>
<td>Path of the share on the server</td>
</tr>
<tr>
<td></td>
<td>comment = Share Foobar on %L from %m</td>
<td>Describe the share to users</td>
</tr>
<tr>
<td></td>
<td>browsable = yes</td>
<td>Show the share when browsing</td>
</tr>
<tr>
<td></td>
<td>writeable = yes</td>
<td>Allow the users to write in this share</td>
</tr>
<tr>
<td></td>
<td>valid users = jdoe, kgreen, +geeks</td>
<td>Allow access only to users “jdoe” and “kgreen”, and to local group “geeks”</td>
</tr>
<tr>
<td></td>
<td>invalid users = csmit</td>
<td>Deny access to user “csmit”</td>
</tr>
<tr>
<td></td>
<td>read list = bcameron</td>
<td>Allow read-only access to user “bcameron”</td>
</tr>
<tr>
<td></td>
<td>write list = fcastle</td>
<td>Allow read-write access to user “fcastle”</td>
</tr>
</tbody>
</table>
### Samba configuration

#### User-level authentication

```
[global]
  security = user

guest account = nobody

map to guest = Never
```

Set up user-level authentication

Map the guest account to the system user `nobody` (default)

Specify how incoming requests are mapped to the guest account:
- **Bad User**: redirect from an invalid user to guest account on server
- **Bad Password**: redirect from an invalid password to guest account on server
- **Never**: reject unauthenticated users

#### Server-level authentication

```
[global]
  security = server

password server = srv1 srv2
```

Set up server-level authentication

Authenticate to server `srv1`, or to server `srv2` if the first one is unavailable

#### Domain-level authentication

```
[global]
  security = ADS

realm = KR0_REALM
```

Set up domain-level authentication as an Active Directory member server

Join the specified realm.

Kerberos must be installed and an administrator account must be created:
```
net ads join -U Administrator\%
```

#### Share-level authentication

```
[global]
  security = share

[foobar]
  path = /foobar

  username = user

  only user = yes
```

Define a "foobar" share accessible to any user which can supply `user`'s password.

The `user` must be created on the system:
```
useradd -c "Foobar account" -d /tmp -m -s /sbin/nologin user
```

... and added to the Samba password file:
```
smbpasswd -a user
```

---

### Samba macros

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%S</td>
<td>Username</td>
</tr>
<tr>
<td>%U</td>
<td>Session username i.e. the username that the client requested, not necessarily the same as the one the client obtained</td>
</tr>
<tr>
<td>%G</td>
<td>Primary group of session username</td>
</tr>
<tr>
<td>%h</td>
<td>Samba server hostname</td>
</tr>
<tr>
<td>%m</td>
<td>NetBIOS name of the server</td>
</tr>
<tr>
<td>%M</td>
<td>Client hostname</td>
</tr>
<tr>
<td>%l</td>
<td>NetBIOS name of the client</td>
</tr>
<tr>
<td>%d</td>
<td>Process ID of the current server process</td>
</tr>
<tr>
<td>%A</td>
<td>Architecture of remote machine</td>
</tr>
<tr>
<td>%I</td>
<td>IP address of client machine</td>
</tr>
<tr>
<td>%i</td>
<td>Local IP address to which a client connected</td>
</tr>
<tr>
<td>%t</td>
<td>Current date and time</td>
</tr>
<tr>
<td>%D</td>
<td>Domain or workgroup of the current user</td>
</tr>
<tr>
<td>%W</td>
<td>Winbind separator</td>
</tr>
<tr>
<td>%${var}</td>
<td>Value of the environment variable <code>var</code></td>
</tr>
</tbody>
</table>

These macros are applied only to configuration options used once a connection has been established:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%S</td>
<td>Name of the current service, if any</td>
</tr>
<tr>
<td>%P</td>
<td>Root directory of the current service, if any</td>
</tr>
<tr>
<td>%U</td>
<td>Username of the current service, if any</td>
</tr>
<tr>
<td>%g</td>
<td>Primary group name of username</td>
</tr>
<tr>
<td>%H</td>
<td>Home directory of username</td>
</tr>
<tr>
<td>%N</td>
<td>Name of the NIS home directory server as obtained from the NIS <code>auto.map</code> entry. Same as %N if Samba was not compiled with the --with-automount option</td>
</tr>
<tr>
<td>%p</td>
<td>Path of service's home directory as obtained from the NIS <code>auto.map</code> entry. The NIS <code>auto.map</code> entry is split up as %N:%p</td>
</tr>
</tbody>
</table>
# Samba - setup

This procedure allows sharing on read-write the local directory `/smbshare` on server 10.1.1.1 to client 10.2.2.2.

## Server setup:

1. Create the group for write access to the share
   ```
   groupadd -r geeks
   ```
2. Create the user and assign it to the group
   ```
   useradd -G geeks jdoe
   ```
3. Add the user to Samba.
   You will be prompted to enter a password
   ```
   smbpasswd -a jdoe
   ```
4. Assign correct ownership to the share
   ```
   chgrp geeks /smbshare
   ```
5. Set the SGID bit to the share
   ```
   chmod 2775 /smbshare
   ```
6. Set the correct SELinux label to the share
   ```
   semanage fcontext -a -t samba_share_t '/smbshare'
   restorecon -FR /smbshare
   ```
7. Enable the SELinux boolean for write access to the share
   ```
   setsebool -P samba_export_all_rw=on
   ```
8. Add a section for the share on `/etc/samba/smb.conf`:
   ```
   [smbshare]
   path = /smbshare
   hosts allow = 10.2.2.2
   write list = @geeks
   ```
9. Ensure that the `smb` and `nmb` services are running

## Client setup:

1. Add an entry to `/etc/fstab` to mount the Samba share device automatically:
   ```
   //10.1.1.1/smbshare /mountpoint cifs username=jdoe,password=s3cr3t  0 0
   ```

## Client multiuser setup:

1. Add an entry to `/etc/fstab` to mount the Samba share device automatically in multiuser mode:
   ```
   //10.1.1.1/smbshare /mountpoint cifs username=jdoe,password=s3cr3t,multiuser,sec=ntlmssp  0 0
   ```
2. Login as another user (there must be a matching Samba user on the Samba server 10.1.1.1)
   ```
   su - ksmith
   ```
3. Store the Samba username and password in the kernel keyring for the current session
   ```
   cifscreds add 10.1.1.1
   ```
A Network File System (NFS) server makes filesystems available to remote clients for mounting.

NFS requires the portmapper to map incoming TCP/IP connections to the appropriate NFS RPC calls. Some Linux distributions use rpcbind instead of the portmapper.

For security reasons, the TCP Wrapper should be configured to limit access to the portmapper to NFS clients only:

file /etc/hosts.deny should contain  portmap: ALL

file /etc/hosts.allow should contain  portmap: IP_addresses_of_clients

NFS handles user permissions across systems by considering users with same UID and username as the same user. Group permission is evaluated similarly, by GID and groupname.

NFS daemons

rpc.nfsd
rpc.mountd
rpc.lockd
rpc.statd

/etc/exports
List of the filesystems to be exported (via the command exportfs)
/var/lib/nfs/xtab
List of exported filesystems, maintained by exportfs
/proc/fs/nfs/exports
Kernel export table (can be examined via the command cat)

exportfs -ra
Export or reexport all directories.
When exporting, fills the kernel export table /proc/fs/nfs/exports.
When reexporting, removes the entries in /var/lib/nfs/xtab that are deleted from /etc/exports (therefore synchronizing the two files), and removes the entries from /proc/fs/nfs/exports that are no longer valid.

exportfs -ua
Unexport all directories.
Removes from /proc/fs/nfs/exports the entries that are listed in /var/lib/nfs/xtab, and clears the latter file.

mount -t nfs nfsserver:/share /usr
Command to be run on a client to mount locally a remote NFS share.
NFS shares accessed frequently should be added to /etc/fstab e.g. nfsserver:/share /usr nfs intr 0 0

showmount
Show the remote client hosts currently having active mounts
showmount --directories
Show the directories currently mounted by a remote client host
showmount --exports
Show the filesystems currently exported i.e. the active export list
showmount --all
Show both remote client hosts and directories
showmount -e nfsserver
Show the shares a NFS server has available for mounting

rpcinfo -p nfsserver
Probe the portmapper on a NFS server and display the list of all registered RPC services there
rpcinfo -t nfsserver nfs
Test a NFS connection by sending a null pseudo request (using TCP)
rpcinfo -u nfsserver nfs
Test a NFS connection by sending a null pseudo request (using UDP)

nfsstat
Display NFS/RPC client/server statistics.

<table>
<thead>
<tr>
<th>Option</th>
<th>NFS</th>
<th>RPC</th>
<th>both</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>-sn</td>
<td>-sr</td>
<td>-s</td>
</tr>
<tr>
<td>client</td>
<td>-cn</td>
<td>-cr</td>
<td>-c</td>
</tr>
<tr>
<td>both</td>
<td>-n</td>
<td>-r</td>
<td>-nr</td>
</tr>
</tbody>
</table>

Options:
### NFS - export table

<table>
<thead>
<tr>
<th>filesystem</th>
<th>Filesystem on the NFS server to be exported to clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>client identification</td>
<td>Client systems permitted to access the exported directory. Can be specified by hostname, IP address, wildcard, subnet, or @NIS workgroup. Multiple client systems can be listed, and each one can have different options</td>
</tr>
<tr>
<td>client options</td>
<td>ro: Read-only access (default)</td>
</tr>
<tr>
<td></td>
<td>rw: Read and write access. The client might choose to mount read-only anyway</td>
</tr>
<tr>
<td></td>
<td>sync: Reply to requests only after the changes made by these requests have been committed to stable storage</td>
</tr>
<tr>
<td></td>
<td>async: Reply to requests without waiting that changes are committed to stable storage. Improves performances but might cause loss or corruption of data if server crashes</td>
</tr>
<tr>
<td></td>
<td>root_squash: Requests by user root on client will be done as user nobody on server (default)</td>
</tr>
<tr>
<td></td>
<td>no_root_squash: Requests by user root on client will be done as same user root on server</td>
</tr>
<tr>
<td></td>
<td>all_squash: Requests by a non-root user on client will be done as user nobody on server</td>
</tr>
<tr>
<td></td>
<td>no_all_squash: Requests by a non-root user on client will be attempted as same user on server (default)</td>
</tr>
</tbody>
</table>

### NFS mount options

<table>
<thead>
<tr>
<th>Mount option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsize=nnn</td>
<td>Size for read transfers (from server to client)</td>
</tr>
<tr>
<td>wsize=nnn</td>
<td>Size for write transfers (from client to server)</td>
</tr>
<tr>
<td>nfsvers=n</td>
<td>Use NFS version n for transport</td>
</tr>
<tr>
<td>retry=n</td>
<td>Keep retrying a mount attempt for n minutes before giving up</td>
</tr>
<tr>
<td>timeo=n</td>
<td>A mount attempt times out after n tenths of a second</td>
</tr>
<tr>
<td>intr</td>
<td>User can interrupt a mount attempt</td>
</tr>
<tr>
<td>nonintr</td>
<td>User cannot interrupt a mount attempt (default)</td>
</tr>
<tr>
<td>hard</td>
<td>The system will try a mount indefinitely (default)</td>
</tr>
<tr>
<td>soft</td>
<td>The system will try a mount until an RPC timeout occurs</td>
</tr>
<tr>
<td>bg</td>
<td>Try a mount in the foreground; all retries occur in the background</td>
</tr>
<tr>
<td>fg</td>
<td>All mount attempts occur in the background (default)</td>
</tr>
<tr>
<td>tcp</td>
<td>Connect using TCP</td>
</tr>
<tr>
<td>udp</td>
<td>Connect using UDP</td>
</tr>
<tr>
<td>sec=krb5p</td>
<td>Use Kerberos to encrypt all requests between client and server</td>
</tr>
<tr>
<td>v4.2</td>
<td>Enable NFS v4.2, which allows the server to export the SELinux context</td>
</tr>
</tbody>
</table>
**NFS setup**

This procedure allows sharing on read-write the local directory `/nfsshare` on server 10.1.1.1 to client 10.2.2.2.

**Server setup:**
1. Ensure that the `nfs-server` service is running
2. Change ownership of the share
   
   ```shell
   chown nfsnobody /nfsshare
   ```
3. Add an entry for the share on `/etc/exports`:
   
   `/nfsshare  10.2.2.2(rw)`
4. Reload the exports file
   
   ```shell
   exportfs -r
   ```

**Client setup:**
1. Add an entry to `/etc/fstab` to mount the NFS share device automatically:

   ```text
   10.1.1.1:/nfsshare  /mountpoint  nfs  defaults  0  0
   ```

---

**Secure NFS setup**

This procedure allows sharing on read-write the local directory `/nfsshare` on server 10.1.1.1 to client 10.2.2.2, securely with Kerberos enabled.

**Server setup:**
1. Install the appropriate server keytab on `/etc/krb5.keytab`
2. Ensure that the `nfs-secure-server` service is running
3. Change ownership of the share
   
   ```shell
   chown nfsnobody /nfsshare
   ```
4. Add an entry for the share on `/etc/exports`:
   
   `/nfsshare  10.2.2.2(sec=krb5p,rw)`
5. Reload the exports file
   
   ```shell
   exportfs -r
   ```

**Client setup:**
1. Install the appropriate client keytab on `/etc/krb5.keytab`
2. Ensure that the `nfs-secure` service is running
3. Add an entry to `/etc/fstab` to mount the NFS share device automatically:

   ```text
   10.1.1.1:/nfsshare  /mountpoint  nfs  defaults,sec=krb5p  0  0
   ```
iSCSI (Internet Small Computer System Interface) is a network protocol that allows emulating an SCSI local storage device over a TCP/IP network. By default it uses TCP port 3260.

An iSCSI server can use a local block device (physical or virtual disk, disk partition, or Logical Volume), a file, a physical SCSI device, or a ramdisk as the underlying storage resource (backstore) and make it available by assigning it a LUN (Logical Unit Number). An iSCSI server provides one or more targets, each of which presents one or more LUNs and is able to accept connections from an iSCSI client (initiator).

Targets and initiators are called nodes and are identified by a unique IQN (iSCSI Qualified Name) e.g. iqn.2017-11.org.example.subdomain:foo:bar. The IP address and port of a node is called a portal.

A target accepts connections from an initiator via a TPG (Target Portal Group) i.e. its IP address and port. A TPG may have in place an ACL so to accept connections only from a specific initiator's IQN.

targetcli

Target configurator (server side). Can be used as a command line tool or as an interactive shell. Configuration is saved to /etc/target/saveconfig.json

iscsiadm

Administration tool for iSCSI devices (client side)
### iSCSI setup

This procedure makes available the local disk `/dev/sbd` on server 10.1.1.1 to the client having IQN `iqn.2017-11.org.example:client`.

#### Server (target) setup:

1. Ensure that the `targetcli` service is running
2. Enter the `targetcli` shell
3. Create a backstore
4. Create a IQN for the target. This automatically creates a TPG for the IQN
5. On the TPG, create an ACL to allow connections from the initiator with a specific IQN
6. On the TPG, create a LUN for the backstore
7. On the TPG, create a portal listening from the server’s IP address
8. Verify the configuration
9. Exit the targetcli shell.

#### Client (initiator) setup:

1. Set the correct initiator IQN in the file `/etc/iscsi/initiatorname.iscsi`:
   ```
   InitiatorName=iqn.2017-11.org.example:client
   ```
2. Ensure that the `iscsi` service is running
3. Discover the iSCSI target(s) provided by the portal. This echoes the target(s) IQN found
4. Login to the target IQN found
5. Add an entry to `/etc/Fstab` to mount the iSCSI device automatically:

```bash
UUID=nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn /mountpoint fs_type _netdev 0 0
```
DHCP (Dynamic Host Configuration Protocol) is a protocol for network management that automatically provides a requesting host with an IP address and other network configuration parameters. It is based on BOOTP (Bootstrap Protocol). A DHCP server listens for requests on UDP port 67 and answers to UDP port 68. The assignment of an IP address to a host is done through a sequence of DHCP messages initiated by the client host, which is (for DHCPv4): DHCP Discover, DHCP Offer, DHCP Request, and then DHCP ACK.

Because DHCP Discover messages are broadcast and therefore not routed outside a LAN, a DHCP relay agent is necessary for those clients situated outside the DHCP server’s LAN. The DHCP relay agent listens to DHCP Discover messages and relays them in unicast to the DHCP server.

/etc/dhcpd.conf: Configuration file for the DHCP server
/etc/sysconfig/dhcprelay (SUSE): Configuration file for the DHCP relay agent
/var/lib/dhcpd/dhcpd.leases: DHCP current leases

<table>
<thead>
<tr>
<th>/etc/dhcpd.conf</th>
<th>DHCP server configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>option domain-name-servers 10.2.2.2;</td>
<td>Global parameters for DNS, mail, NTP, and news servers specification</td>
</tr>
<tr>
<td>option smtp-servers 10.3.3.3;</td>
<td>Definition of a network</td>
</tr>
<tr>
<td>option pop-servers 10.4.4.4;</td>
<td>Time, in seconds, that will be assigned to a lease if a client does not ask for a specific expiration time</td>
</tr>
<tr>
<td>option time-servers 10.5.5.5;</td>
<td>Maximum time, in seconds, that can be assigned to a lease if a client asks for a specific expiration time</td>
</tr>
<tr>
<td>option nntp-servers 10.6.6.6;</td>
<td>Definition of different subnets in the network, with specification of different ranges of IP addresses that will be leased to clients depending on the client’s subnet</td>
</tr>
<tr>
<td>shared-network geek-net {</td>
<td>Definition of a group</td>
</tr>
<tr>
<td>default-lease-time 86400;</td>
<td>Definition of different hosts to whom static IP addresses will be assigned to, depending on their MAC address</td>
</tr>
<tr>
<td>max-lease-time 172800;</td>
<td>}</td>
</tr>
<tr>
<td>option routers 10.0.3.252;</td>
<td></td>
</tr>
<tr>
<td>option broadcast-address 10.0.3.255;</td>
<td></td>
</tr>
<tr>
<td>subnet 10.0.3.0 netmask 255.255.255.128 {</td>
<td></td>
</tr>
<tr>
<td>range 10.0.3.1 10.0.3.101;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>subnet 10.0.3.128 netmask 255.255.255.128 {</td>
<td></td>
</tr>
<tr>
<td>range 10.0.3.129 10.0.3.229;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>group {</td>
<td></td>
</tr>
<tr>
<td>option routers 10.0.17.252;</td>
<td></td>
</tr>
<tr>
<td>option broadcast-address 10.0.17.255;</td>
<td></td>
</tr>
<tr>
<td>netmask 255.255.255.0;</td>
<td></td>
</tr>
<tr>
<td>host linuxbox1 {</td>
<td></td>
</tr>
<tr>
<td>hardware ethernet AA:BB:CC:DD:EE:FF;</td>
<td></td>
</tr>
<tr>
<td>fixed-address 10.0.17.42;</td>
<td></td>
</tr>
<tr>
<td>option host-name &quot;linuxbox1&quot;;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>host linuxbox2 {</td>
<td></td>
</tr>
<tr>
<td>fixed-address 10.0.17.66;</td>
<td></td>
</tr>
<tr>
<td>option host-name &quot;linuxbox2&quot;;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>
### DHCP - message types

<table>
<thead>
<tr>
<th>DHCPv4 message</th>
<th>DHCPv6 message</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP Discover</td>
<td>Solicit</td>
</tr>
<tr>
<td>DHCP Offer</td>
<td>Advertise</td>
</tr>
<tr>
<td><strong>DHCP Request</strong></td>
<td>Request</td>
</tr>
<tr>
<td></td>
<td>Renew</td>
</tr>
<tr>
<td></td>
<td>Rebind</td>
</tr>
<tr>
<td></td>
<td>Confirm</td>
</tr>
<tr>
<td>DHCP ACK</td>
<td>Reply</td>
</tr>
<tr>
<td>DHCP Release</td>
<td>Release</td>
</tr>
<tr>
<td>DHCP Decline</td>
<td>Decline</td>
</tr>
<tr>
<td></td>
<td>Reconfigure</td>
</tr>
<tr>
<td>DHCP Inform</td>
<td>Information-Request</td>
</tr>
<tr>
<td></td>
<td>Relay-Forward</td>
</tr>
<tr>
<td></td>
<td>Relay-Reply</td>
</tr>
<tr>
<td>DHCP NAK</td>
<td></td>
</tr>
</tbody>
</table>
PAM (Pluggable Authentication Modules) is an abstraction layer that allows applications to use authentication methods while being implementation-agnostic.

/etc/pam.d/service
/etc/pam.conf (obsolete)

PAM configuration for service
PAM configuration for all services

ldd /usr/sbin/service | grep libpam

Check if service is enabled to use PAM

<table>
<thead>
<tr>
<th>/etc/pam.d/service</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| type | auth | Authentication module to verify user identity and group membership |
|      | account | Authorization module to determine user's right to access a resource (other than their identity) |
|      | password | Module to update a user's authentication credentials |
|      | session | Module (run at end and beginning of a user session) to set up the user environment |

| control | optional | Module is not critical to the success or failure of service |
|         | sufficient | If this module successes, and no previous module has failed, module stack processing ends successfully. If this module fails, it is non-fatal and processing of the stack continues |
|         | required  | If this module fails, processing of the stack continues until the end, and service fails |
|         | requisite | If this module fails, service fails and control returns to the application that invoked service |
|         | include   | Include modules from another PAM service file |

| module | PAM module and its options, e.g.: |
|        | pam_unix.so | Standard UNIX authentication module via /etc/passwd and /etc/shadow |
|        | pam_nis.so  | Module for authentication via NIS |
|        | pam_ldap.so | Module for authentication via LDAP |
|        | pam_fshadow.so | Module for authentication against an alternative shadow passwords file |
|        | pam_cracklib.so | Module for password strength policies (e.g. length, case, max number of retries) |
|        | pam_limits.so | Module for system policies and system resource usage limits |
|        | pam_listfile.so | Module to deny or allow the service based on an arbitrary text file |
LDAP (Lightweight Directory Access Protocol) is a simplified version of the X.500 standard and uses TCP port 389. LDAP allows organizing hierarchically a database of entries, each one of which is identified by a unique DN (Distinguished Name). Each DN has a set of attributes, and each attribute has a value; an attribute may appear multiple times. Special attributes called objectClass define which attributes are allowed and which are required, and determine the schema of the LDAP.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute with value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>cn</td>
<td>cn: John Doe</td>
<td>Common Name</td>
</tr>
<tr>
<td>dc</td>
<td>dc=example,dc=org</td>
<td>Domain Component</td>
</tr>
<tr>
<td>givenName</td>
<td>givenName: John</td>
<td>First name</td>
</tr>
<tr>
<td>sn</td>
<td>sn: Doe</td>
<td>Surname</td>
</tr>
<tr>
<td>mail</td>
<td>mail: <a href="mailto:jdoe@example.org">jdoe@example.org</a></td>
<td>Email address</td>
</tr>
<tr>
<td>telephoneNumber</td>
<td>telephoneNumber: +1 555 1234 567</td>
<td>Telephone number</td>
</tr>
<tr>
<td>uid</td>
<td>uid: jdoe</td>
<td>User ID</td>
</tr>
<tr>
<td>c</td>
<td>c: US</td>
<td>Country code</td>
</tr>
<tr>
<td>l</td>
<td>l: San Francisco</td>
<td>Locality</td>
</tr>
<tr>
<td>st</td>
<td>st: California</td>
<td>State or province</td>
</tr>
<tr>
<td>street</td>
<td>street: 42, Penguin Road</td>
<td>Street</td>
</tr>
<tr>
<td>o</td>
<td>o: The Example Foundation</td>
<td>Organization</td>
</tr>
<tr>
<td>ou</td>
<td>ou: IT Dept</td>
<td>Organizational Unit</td>
</tr>
<tr>
<td>manager</td>
<td>manager: cn=Kim Green,dc=example,dc=org</td>
<td>Manager</td>
</tr>
</tbody>
</table>

**LDIF (LDAP Data Interchange Format)**

```ldif
dn: cn=John Doe, dc=example, dc=org
changetype: modify
replace: mail
mail: johndoe@otherexample.com
-
add: jpegPhoto
jpegPhoto:< file://tmp/jdoe.jpg
-
delete: description
```

This LDIF file will change the email address of user "jdoe", add a picture, and delete the description attribute for the entry.
All the LDAP commands below accept the following arguments, plus some extra arguments which are command-dependent.

- **-H ldap://** Connect to the specified LDAP server
- **-H ldapi://** Connect to the localhost LDAP server using IPC instead of a network socket
- **-D binddn** Bind (authenticate) to the LDAP server as the specified DN
- **-w password** Authenticate with the specified `password`
- **-W** Prompt for authentication
- **-x** Use simple authentication instead of SASL
- **-v** Use verbose mode for output

**ldapsearch args**

Query an LDAP server and return the output in LDIF

- **-b base** Start searching from `base`
- **-z n** Retrieve at maximum `n` entries as result
- **-LLL** Terse output. Outputs the result in LDIFv1, does not print comments, and omits the LDIF version number
- **filter** Search filter. If not specified, uses the default filter `(objectClass=*)`
- **attributes** Attributes to return. If not specified, returns all attributes

**ldapmodify args**

Modify an LDAP entry

**ldapadd args**

Add an LDAP entry

**ldapmodify -a args**

Delete an LDAP entry

- **-f file.ldif** Modify the entry according to the LDIF file

**ldapdelete args**

Delete an LDAP entry

- **-f file.ldif** Modify the entry according to the LDIF file

**ldappasswd args**

Change the password of an LDAP entry

- **-s password** Set the new password as `password`
- **-S** Prompt for the new password

Example commands:

```
ldapsearch -H ldap://ldap.example.org -s base "(sn=Doe)"
ldapmodify -b -r -f file.ldif
ldapadd -h ldap.example.org -D "cn=Admin,dc=example,dc=org" -W -f file.ldif
ldapdelete -h ldap.example.org -D "cn=Admin,dc=example,dc=org" -W -x "uid=jdoe,dc=example,dc=org"
ldappasswd -h ldap.example.org -D "cn=Admin,dc=example,dc=org" -W -x "uid=jdoe,ou=IT Dept,dc=example,dc=org"
```

Query a LDAP server for entries in the OU "people" whose surname is "Doe"; print common name, surname, and telephone number of the entries found

Modify an entry according to the LDIF file specified

Authenticateating as "Admin", add an entry by adding the content of the specified LDIF file to the directory

Authenticateating as "Admin", delete the user "jdoe"

Authenticateating as "Admin" on example.org, change the password of user "jdoe" in the OU "IT Dept"
OpenLDAP is an open source implementation of LDAP, and was initially developed together with the LDAP protocol. Its related service is slapd, the Standalone OpenLDAP daemon. sssd, the System Security Services Daemon, can be used to provide access to OpenLDAP as an authentication and identity provider.

/var/lib/ldap/

/etc/openldap/slapd.conf
/usr/local/etc/openldap/slapd.conf
/usr/local/etc/openldap/slapd.d/

Files constituting the OpenLDAP database
OpenLDAP configuration file (deprecated)
From v2.3 onwards, directory containing the LDIF database that stores the OpenLDAP configuration. These LDIF files must not be edited by hand

slapcat -b cn=config
ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config

Show the OpenLDAP configuration
Verify that the OpenLDAP configuration is correct

slaptest -u

Dump the contents of an OpenLDAP database to an LDIF file

slapcat -l file.ldif
slapadd -l file.ldif
slapindex

Import an OpenLDAP database from an LDIF file
Regenerate OpenLDAP’s database indexes

yum install openldap openldap-clients authconfig \ ssd nss-pam-ldapd authconfig-gtk (RHEL 7)
yum install openldap openldap-clients authconfig \ ssd nss-pam-ldapd authconfig-gtk (RHEL 7)

Set up the LDAP client to connect to a ldapserver. This will update the configuration files /etc/sssd/sssd.conf and /etc/openldap/ldap.conf
Set up LDAP client authentication via sssd

authconfig --enableldap --enableldapauth \ --enableldaps --ldapsupport=ldap://ldapserver: \ --ldapbasedn="dc=example,dc=org" \ --enableszept --update (RHEL 7)
authselect select sssd --force (RHEL 8)

authconfig-gtk
system-config-authentication

Install the OpenLDAP client
OpenLDAP configuration GUI

getent group groupname

Get entries about groupname from NSS libraries
Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies.

SELinux implements a Mandatory Access Control framework that allows the definition of fine-grained permissions for how subjects (i.e. processes) access objects (i.e. other processes, files, devices, ports, sockets); this improves security with respect to the traditional Discretionary Access Control, which defines accesses based on users and groups.

Processes, files, and users have a security context structured as user:role:type:level e.g. unconfined_u:object_r:user_home_t:s0. The third field defines a type for files or a domain for processes. The decisions SELinux takes about allowing or disallowing access are stored in the AVC (Access Vector Cache).

SELinux creates a pseudo filesystem (SELinuxfs) containing commands used by the kernel for its operations; this filesystem is usually mounted on /selinux/ or /sys/fs/selinux/.

```
setenforce 0
echo 0 > /selinuxfs/enforce
```

Enter permissive mode (SELinux must be enabled)

```
setenforce 1
echo 1 > /selinuxfs/enforce
```

Enter enforcing mode (SELinux must be enabled)

```
getenforce
cat /selinuxfs/enforce
```

Display current mode

```
sestatus -v
```

Show SELinux mode, SELinuxfs mount point, etc.

SELinux state can be configured permanently in /etc/selinux/config (symlinked in /etc/sysconfig/selinux):

<table>
<thead>
<tr>
<th>mode</th>
<th>SELINUX=</th>
<th>SELinux fully enforces security policies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enforcing</td>
<td>SELinux does not enforce security policies, but logs violations</td>
</tr>
<tr>
<td></td>
<td>permissive</td>
<td>SELinux security policies are disabled</td>
</tr>
<tr>
<td></td>
<td>disabled</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>policy</th>
<th>SELINUXTYPE=</th>
<th>SELinux protects targeted daemons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>targeted</td>
<td>(up to RHEL 6) SELinux fully protects the system</td>
</tr>
<tr>
<td></td>
<td>strict</td>
<td>(RHEL 7 and later) SELinux only protects selected processes</td>
</tr>
<tr>
<td></td>
<td>minimum</td>
<td>(RHEL 7 and later) Multi Level Security protection</td>
</tr>
<tr>
<td></td>
<td>mls</td>
<td></td>
</tr>
</tbody>
</table>

```
ls -Z
```

List files and their security context. The security context of a file is stored in its extended attributes

```
cp --preserve=context file file2
```

Copy a file and its context. By default, the cp command will create a new SELinux file context

```
ps -eZ
```

List processes and their security context

```
tar --selinux otherargs
star -xattr -H=exustar otherargs
```

Create or extract archives that retain the security context of the original files
semange

Manage SELinux policies

```bash
semange fcontext -l
```
List files and their assigned SELinux labels

```bash
semange fcontext -a -t label file
```
Assign the SELinux label to file. It is then necessary to apply the label via `restorecon -f file`

```bash
semange login -l
```
List mappings between users and SELinux users

```bash
semange port -l
```
List port numbers and their assigned SELinux type definitions

```bash
semange port -a -t portlabel -p tcp n
```
Assign the SELinux portlabel to TCP port n

```bash
semange port -d -t http_port_t -p tcp 8888
```
Remove the binding of http_port_t port label to TCP 8888

```bash
semange port -m -t http_cache_port_t -p tcp 8888
```
Modify the port label bound to TCP 8888

```bash
semange permissive -a auditd_t
```
Add auditd_t to the list of permissive types/domains. In this case, SELinux allows the auditd daemon all access while logging its AVC violations

```bash
semange permissive -d auditd_t
```
Delete auditd_t from the list of permissive types/domains

```bash
semange permissive -l
```
List all permissive types/domains

sepolicy

Inspect a SELinux policy

```bash
sepolicy manpage -a -p /usr/local/man/man8 && mandb
```
Generate all SELinux policy manpages

seinfo

Query the components of a SELinux policy

```bash
chcon context file
```
Change the security context of file to the specified context

```bash
chcon --reference=file0 file
```
Change the security context of file to be the same as file0

```bash
restorecon -f file
```
Restore the security context of file to the system default

```bash
getsebool boolean
```
Get the value of a SELinux boolean

```bash
setsebool boolean=value
```
Set the value of a SELinux boolean
Logfile containing AVC denials, if `auditd` is running (default). AVC denials can also be seen via `dmesg`

Logfile containing AVC denials, if `rsyslogd` is running

```
/var/log/audit/audit.log
/var/log/messages
```

`sealert -a logfile`

Analyze a SELinux logfile and display verbosely SELinux policy violations. SELinux violation events are logged as `type=AVC msg=audit(timest.amp:id): avc: denied (...)`

```
grep timest.amp:id logfile | audit2why
```

Diagnostic a specific AVC denial event entry (identified by a timestamp and an id) from a SELinux logfile

```
audit2why -d
```

Read AVC violations from the output of `dmesg`

```
ausearch -a id
```

Query the SELinux log for event id

```
audit2allow -i inputfile -M module
```

Generate a loadable module containing the appropriate SELinux policy from a denied operation stored in `inputfile`

```
ausearch -c '({exe})' --raw | audit2allow -M module
```

Generate a loadable module to allow access on an executable which caused an AVC violation. This module will then need to be installed via `semodule -i module.pp`

```
semodule -l
```

List installed SELinux policy modules

```
semodule -X n -i module.pp
```

Install a SELinux policy module at priority n. Installed modules persist after reboot. Module files have usually the suffix `.pp` (policy package)

```
semodule -X n -r module
```

Remove a SELinux policy module at priority n. Modules must be removed at the same priority at which they were installed
Kickstart is a method to perform automatic installation and configuration of RHEL machines. This can be done by specifying `inst.ks=hd:/dev/sda:/root/path/ksfile` either as a boot option, or an option to the kernel command in GRUB 2.

```
/root/anaconda-ks.cfg
```

Kickstart file describing the current system. This file is automatically generated during the installation.

```
system-config-kickstart
ksvalidator ksfile
ksverdiff -f RHEL6 -t RHEL7
```

- GUI tool to create a Kickstart file
- Check the validity of a Kickstart file
- Show the differences in the Kickstart syntax between RHEL 6 and RHEL 7
**Red Hat Satellite** is a system management software platform that allows provisioning and configuration of RHEL machines. Repository content is provided via Red Hat Subscription Management (RHSM).

Satellite 5 was based on Spacewalk, an open source system management software for Linux machines. Satellite 6 is a complete overhaul of it and is composed of:

- **Foreman**, an open source lifecycle management tool able to provision servers via Kickstart and Puppet;
- **Katello**, a tool that handles Red Hat repository management (via the **Pulp** service) and subscription management (via the **Candlepin** service).

All these components above need a PostgreSQL database, except Pulp which needs a MongoDB database.

As a separate component, **Capsule** servers act as a proxy for many of the main Satellite functions e.g. repository storage. A Capsule is also integrated in each Satellite server.

```
subscription-manager register
subscription-manager attach
foreman-maintain service list
foreman-maintain service status
foreman-maintain service start
foreman-maintain service stop
foreman-maintain service restart
foreman-maintain backup
foreman-rake command:option
hammer
pulp-admin-client
virt-who
foreman-debug
sosreport
citellus.py sosreportfile
```

- **Register a system to the RHSM portal**
- **Attach a RHSM subscription to a registered system**
- **List all Satellite services**
- **Display status or start, stop, restart all Satellite services.**
  - Performed via **systemctl**
- **Make a backup of Satellite**
- **Perform various administrative tasks**
- **CLI tool for Foreman**
- **Tool to administer the Pulp server**
- **Agent for scanning and reporting virtual guest IDs and hypervisors to a Satellite server**
- **Collect Satellite configuration, log, and backend data for debug purposes**
- **Collect diagnostic and configuration data for technical support**
- **Perform some automated checks for troubleshooting a system**
KVM (Kernel-based Virtual Machine) is a virtualization infrastructure for the Linux kernel that allows it to function as a hypervisor.

/etc/libvirt/qemu/

Directory containing the XML files that define VMs properties.

/libvirtd must be restarted after modifying an XML file

/var/lib/libvirt/

Directory containing files related to the VMs

virt-manager

KVM GUI

virt-install --prompt

Interactive command-line program to create a VM

Create a VM with 2 Gb of RAM, specifying path of virtual disk, location of installation files, and (as extra argument) the Kickstart configuration to use

virt-install -n vmname -r 2048 \ --disk path=/var/lib/libvirt/images/vmname.img \ -l /root/vmstuff/inst/ \ -x "ks=/root/vmstuff/kickstart.cfg"

virt-clone --prompt

Interactive command-line program to clone a VM. A VM must be shut off or paused before it can be cloned

Clone a VM

virt-clone -o vmname -n vmclonename

virsh

Interface for VM management

virsh list --all

List all VMs present on the system

virsh start vmname

Start a VM

virsh destroy vmname

Brutally shut down a VM

virsh shutdown vmname

Gracefully shut down a VM

virsh autostart vmname

Set a VM to be automatically started when the system boots. Done by symlinking the VM to /etc/libvirt/qemu/autostart/

virsh autostart --disable vmname

Disable the autostart of a VM at system boot

virsh edit vmname

Edit the XML file defining a VM’s properties

virt-what

Detect whether the current machine is a VM
Git is an open source version control system with a small footprint and very high performances. A Git directory is a complete repository with full history and version tracking abilities, independent of any remote repository. Git commits are identified by a 40-hex-digits hash number, usually shortened to 7 digits, or even less if unambiguous.

```
git init
Initialize the current directory as a repository

git clone repo
Clone a remote repository. 
repo can be a URL (SSH, HTTP, HTTPS, FTP, FTPS, Git) or a local path e.g. 
ssh://user@example.com:8888/path/to/repo.git
git://example.com:9999/path/to/repo.git
/path/to/repo.git

git checkout branch
Start working into an already existing branch

git checkout -B branch
Create branch and start working into it

git checkout -- file
Discard local changes done to file

git checkout branch file
Copy file from branch to the current branch, and add it to the staging area

git pull
Pull the changes from the remote repository branch to the local branch

git add file
Add file to the staging area (i.e. content staged for the next commit), hence starting to track it

git add .
Add all modified files to the staging area

git rm file
Remove file from the content staged for the next commit

git status
See the status (e.g. files changed but not yet staged) of the current branch

git commit -m "Message"
Commit all staged files in the current branch

git commit -am "Message"
Add all changed files to the staging area in the current branch, and commit them

git merge branch
Merge changes made on branch to the master branch

git push
Push the local commits from the current branch to the remote repository

git push origin branch
Push the local commits from branch to the remote repository

git revert commit
Revert a specific commit

git branch
Show local branches

git branch -r
Show remote branches

git branch -a
Show remote and local branches

git branch -a --contains commit
Show on which branch was done a specific commit number

git branch -d branch
Delete a local branch (which must have been merged in its upstream branch)

git branch -D branch
Delete a local branch (irrespective of its merged status)
```
git diff commit1 commit2  
Show the differences between two commits

git diff branch1 branch2  
Show the differences between two branches

git diff branch1 branch2 file  
Show the differences between two branches for a specific file

git log --all -- file  
Show the commits which involved file, across all branches

git log -p --all -S 'string'  
Show the commits whose added or deleted lines contain a specific word

git grep string 'git show-ref --heads'  
Search for string across all branches' heads (i.e. in the latest content only, and not in all the previous commits)


git config --list  
Get all currently set options and their values in the Git configuration

git config option  
Get the value of option


git config user.name name  
Set your username

git config user.email email  
Set your email address
Vagrant is an open source software that allows building and maintaining lightweight and portable virtual environments for software development. It relies on an underlying virtualization solution e.g. VirtualBox.

vagrant -h
Print the list of commands recognized by Vagrant

vagrant command -h
Print help about the Vagrant command

vagrant init hashicorp/precise64
Initialize the current directory as a specific Vagrant environment (in this case, Ubuntu 12.04 64-bit) by creating a Vagrantfile on it

vagrant up vmname
Start a guest virtual machine and do a first provisioning according to the Vagrantfile

vagrant provision vmname
Provision a virtual machine

vagrant ssh vmname
Connect via SSH to a virtual machine

vagrant halt vmname
Shut down the virtual machine

vagrant destroy vmname
Delete the virtual machine and free any resource allocated to it

vagrant status
Print the status of the virtual machines currently managed by Vagrant

vagrant global-status
Print the status of all Vagrant environments on the system, by reading cached data. Completes quickly but results may be outdated

vagrant global-status --prune
Print the status of all Vagrant environments on the system, after rebuilding the environment information cache. Results are always correct but completion takes longer

The directory containing the Vagrantfile on the host can be accessed on the guest via /vagrant.
Puppet is a software configuration management tool. It is based on a client-server architecture, where a Puppet agent (client, running as root on each managed node) periodically gathers information (facts) about the local node state via the Facter tool, then communicates this information to the Puppet master (server, running as the puppet user and listening on TCP port 8140). The Puppet master then sends back to the Puppet agent a catalog containing the desired configuration for that node. The Puppet agent applies the needed changes so that the node's configuration converges with the desired configuration, and sends back a report to the Puppet master. Puppet changes are idempotent.

Puppet configurations are based on resources (e.g. "package", "service", "file", "user" ...). For each resource, a list of attributes is specified, with the desired value for each attribute. Each resource type is implemented through providers (e.g. yum, rpm, apt, opkg ... for the resource "package"). Resources managed together as a single unit can be grouped into classes; classes are contained in manifests which are files with the .pp extension. Modules are directories containing self-contained pieces of configuration and classes for a specific complex setting, e.g. an Apache webserver or a MySQL server.

/etc/puppet/puppet.conf
/etc/puppetlabs/puppet/puppet.conf

facter

puppet agent

puppet agent --enable
puppet agent --disable "Reason for disabling"
cat $(puppet config print vardir)/state/agent_disabled.lock

puppet agent --noop

puppet --version
puppet agent --version
puppet master --version

puppet module list
puppet resource user username
puppet resource service httpd enable=false
puppet describe user
puppet describe --list
puppet describe user --providers
puppet apply modulename/init.pp
puppet cert operation

Configuration file (Open Source Puppet)
Configuration file (Puppet Enterprise)

Gather the facts about the managed node, returning a list of key-value pairs

Main Puppet client. Retrieves the node's desired configuration from the Puppet master and applies it

Enable the Puppet agent on the node
Disable the Puppet agent on the node
Print the reason why the Puppet agent is currently disabled. If the Puppet agent is enabled instead, the lockfile does not exist

Perform a dry run, displaying the changes that Puppet would have applied without actually applying them

Show version of different Puppet components

List all modules installed in Puppet
Inspect the state of the resource "user" with respect to username
Modify the state of the resource "service" (in this case, disable the HTTP server)
Show information about the resource "user"
List all resource types
Return the list of providers for the resource "user"
Apply a manifest one time only
Manage the SSL certificates used for communications between master and agents
Ansible is an open source tool (provided by Red Hat) for configuration management, software provisioning, and application deployment. It is agentless and connects to the managed machines via SSH pubkey authentication, requiring only OpenSSH and Python to be installed on the managed nodes.

The configuration for a managed node is specified in a playbook, written in YAML. A playbook contains one or more plays to be run in sequence, each of which lists a number of small specific tasks to be run in sequence. Each task calls a module, which describes the desired state of the system and executes the operation to do so (e.g. start the Apache server, copy a file, verify that a package is installed, rename a database table). A module should be idempotent i.e. it should set the system always in the same state regardless of how many times it is run.

Handlers are tasks that are run only via a notification system, when a change is made on a machine. The inventory lists and categorizes all hosts on which tasks have to be executed. It is also possible to define roles to categorize hosts and automatically load variables, tasks, handlers, and other artifacts pertaining to that role.

When a playbook is run, first it collects system and environment information (facts) which is then stored in multiple variables named ansible_varname.

Ansible uses the Jinja template engine to enable access to tests, filters, variables, and dynamic execution. Also available is a set of plugins that expand Ansible's core functionalities.

Ansible Tower is a web console for user-friendly configuration and management of Ansible elements, and includes web services and a REST API.

//etc/ansible/hosts
Inventory file, containing the list of hosts managed by Ansible. Can be in INI or YAML format

```
ansible hosts -m module -a options
ansible hosts -m ansible.builtin.copy -a "src=/path/to/file dest=/tmp/"
ansible hosts -m ansible.builtin.yum -a "name=httpd state=latest"
ansible hosts -a "/sbin/shutdown"
ansible all -m ping
ansible all -m ansible.builtin.setup

ansible-playbook playbook.yml
```
Apply the specified playbook

```
anible-lint playbook.yml
```
Check the syntax of the specified playbook

```
anible-pull
```
Pull a remote copy of Ansible on each managed node and download the playbooks from a source repository. This inverts the default push architecture of Ansible into a pull architecture

<table>
<thead>
<tr>
<th>ansible command line options</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-m module</code></td>
</tr>
<tr>
<td><code>-a options</code></td>
</tr>
<tr>
<td><code>-f n</code></td>
</tr>
<tr>
<td><code>-u user</code></td>
</tr>
<tr>
<td><code>--become</code></td>
</tr>
<tr>
<td><code>--become --ask-become-pass</code></td>
</tr>
<tr>
<td><code>--become</code></td>
</tr>
</tbody>
</table>
### Inventory and variables

<table>
<thead>
<tr>
<th>hosts</th>
<th>group_vars/all</th>
<th>group_vars/dbservers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory file, defining two groups</td>
<td>Variables applied to all host groups</td>
<td>Variables applied to hosts in the &quot;dbservers&quot; group</td>
</tr>
</tbody>
</table>

- **[webservers]**
  - 10.0.1.17
  - 10.0.1.18

- **[dbservers]**
  - 10.0.2.42

<table>
<thead>
<tr>
<th>group_vars/all</th>
<th>group_vars/dbservers</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>httpd_port: 80</td>
<td>mysqlservice: mysqld</td>
</tr>
<tr>
<td>ntpserver: 192.168.0.13</td>
<td>mysql_port: 3306</td>
</tr>
<tr>
<td>repository: <a href="https://foobar.org/repo/website.git">https://foobar.org/repo/website.git</a></td>
<td>dbuser: jdoe</td>
</tr>
<tr>
<td></td>
<td>dbname: mydb</td>
</tr>
<tr>
<td></td>
<td>dbpassword: mys3cr3t</td>
</tr>
</tbody>
</table>

### Main playbook

**lamp.yml**

Playbook that deploys the whole LAMP stack on the site

---

- name: Apply common configuration to all nodes
  hosts: all
  remote_user: root
  roles:
    - common

- name: Deploy and configure webservers
  hosts: webservers
  remote_user: root
  roles:
    - web

- name: Deploy MySQL and configure databases
  hosts: dbservers
  remote_user: root
  roles:
    - db

### Role: common

**roles/common/handlers/main.yml**

Handler for general notifications, called from other plays

---

- name: Restart NTP
  service: ntpd
  state: restarted

**roles/common/tasks/main.yml**

General play run on all nodes

---

- name: Install NTP
  yum:
    name: ntp
    state: present
tags: ntp

- name: Configure NTP
  template:
    src: ntp.conf.j2
    dest: /etc/ntp.conf
tags: ntp
  notify: Restart NTP

- name: Start NTP
  service:
    name: ntpd
    state: started
tags: ntp

**roles/common/templates/ntp.conf.j2**

Jinja template for NTP configuration file

```jinja
driftfile /var/lib/ntp/drift
restrict 127.0.0.1
server {{ ntpserver }}
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```
Role: db

Handler for DB tier notifications
---
- name: Restart MySQL
  service:
    name: mysqld
    state: restarted
- name: Restart iptables
  service:
    name: iptables
    state: restarted

Install MySQL, then create database and database user
---
- name: Install the MySQL package
  yum:
    name: "{{ item }}"
    state: installed
  with_items:
    - mysql-server
    - MySQL-python
- name: Configure MySQL
  template:
    src: my.cnf.j2
    dest: /etc/my.cnf
    notify: Restart MySQL
- name: Start MySQL
  service:
    name: mysqld
    state: started
    enabled: yes
- name: Insert iptables rule for MySQL
  lineinfile:
    dest: /etc/sysconfig/iptables
    state: present
    regexp: "{{ mysql_port }}" insertafter: "^:OUTPUT "
    line: "-A INPUT -p tcp  --dport {{ mysql_port }} -j ACCEPT"
- name: Create database
  mysql_db:
    name: "{{ dbname }}"
    state: present
- name: Create database user and set password
  mysql_user:
    name: "{{ dbuser }}"
    password: "{{ dbpassword }}"
    priv: "*.*:ALL"
    host: "%"
    state: present

Jinja template for MySQL configuration file
roles/db/templates/my.cnf.j2

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0
port={{ mysql_port }}

[mysqld_safe]
log-error=/var/log/mysqlld.log
pid-file=/var/run/mysqlld/mysqlld.pid
```
## Ansible - playbook example part 3

### Role: web

**roles/web/handlers/main.yml**  
Handler for web tier notifications

---

- name: Restart iptables  
service:  
  name: iptables  
  state: restarted

**roles/web/tasks/main.yml**  
Main task, calls the other two playbooks

---

- include: install_httpd.yml  
- include: copy_website.yml

**roles/web/tasks/copy_website.yml**  
Copy the code from the git repository

---

- name: Copy website from repo  
git:  
  repo: "{{ repository }}"  
  dest: /var/www/html/

- name: Create the index.php file  
template:  
  src: index.php.j2  

**roles/web/tasks/install_httpd.yml**  
Install HTTP, PHP, and Git modules

---

- name: Install httpd packages  
yum:  
  name: "{{ item }}"  
  state: present  
  with_items:  
    - httpd  
    - php  
    - php-mysql  
    - git

- name: Insert iptables rule for httpd  
lineinfile:  
  dest: /etc/sysconfig/iptables  
  create: yes  
  state: present  
  regexp: "{{ httpd_port }}"  
  insertafter: "^:OUTPUT "  
  line: "-A INPUT -p tcp --dport {{ httpd_port }} -j ACCEPT"  
notify: Restart iptables

- name: Check that httpd is running  
service:  
  name: httpd  
  state: started  
  enabled: yes

**roles/web/templates/index.php.j2**  
Jinja template for the website root file

```html
<html>
<head>
<title>LAMP stack and website deployed via Ansible</title>
</head>
<body>
  <a href=http://{{ ansible_default_ipv4.address }}/index.html>Homepage</a>

<?php
  echo "Hostname: " . exec('hostname') . "\n";
  echo "Database list: \n";
  {% for host in groups['dbservers'] %}
    $link = mysqli_connect(
      '{{ hostvars[host].ansible_default_ipv4.address }}',
      '{{ hostvars[host].dbuser }}',
      '{{ hostvars[host].dbpassword }}')
      or die(mysqli_connect_error($link));
  {% endfor %}
  while ($r = mysqli_fetch_assoc(mysqli_query($link, "SHOW DATABASES;")))
    {echo $r['Database'] . "\n";}
  
</body>
</html>
```
### HTML 4.01 - components

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;h1&gt;</code> ... <code>&lt;h6&gt;</code></td>
<td><strong>Heading</strong></td>
<td>**align=left</td>
</tr>
<tr>
<td><code>&lt;br&gt;</code></td>
<td><strong>Line break</strong></td>
<td><strong>Line break and carriage return</strong></td>
</tr>
<tr>
<td><code>&lt;hr&gt;</code></td>
<td><strong>Horizontal line</strong></td>
<td>**align=left</td>
</tr>
<tr>
<td><code>&lt;p&gt;</code></td>
<td><strong>Paragraph</strong></td>
<td>**align=left</td>
</tr>
<tr>
<td><code>&lt;div&gt;</code></td>
<td><strong>Section</strong></td>
<td>**align=left</td>
</tr>
<tr>
<td><code>&lt;span&gt;</code></td>
<td><strong>Group</strong></td>
<td><strong>charset=encoding</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**coords=left,top,right,bottom</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>href=url</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>hreflang=language</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>name=section</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**rel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>**shape=rectangle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>**target=_blank</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>type=mimetype</strong></td>
</tr>
<tr>
<td><code>&lt;dl&gt;</code></td>
<td><strong>Definition list</strong></td>
<td><strong>compact=compact</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>start=firstnumber</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**type=A</td>
</tr>
<tr>
<td><code>&lt;ul&gt;</code></td>
<td><strong>Unordered list</strong></td>
<td><strong>compact=compact</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>**type=disc</td>
</tr>
<tr>
<td><code>&lt;li&gt;</code></td>
<td><strong>List item</strong></td>
<td>**type=disc</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>value=itemno</strong></td>
</tr>
</tbody>
</table>

† = deprecated
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;i&gt;</code></td>
<td>Italic</td>
</tr>
<tr>
<td><code>&lt;b&gt;</code></td>
<td>Bold</td>
</tr>
<tr>
<td><code>&lt;s&gt;</code></td>
<td>Strike-through text †</td>
</tr>
<tr>
<td><code>&lt;u&gt;</code></td>
<td>Underlined text †</td>
</tr>
<tr>
<td><code>&lt;big&gt;</code></td>
<td>Bigger</td>
</tr>
<tr>
<td><code>&lt;small&gt;</code></td>
<td>Smaller</td>
</tr>
<tr>
<td><code>&lt;sub&gt;</code></td>
<td>Subscript</td>
</tr>
<tr>
<td><code>&lt;sup&gt;</code></td>
<td>Superscript</td>
</tr>
<tr>
<td><code>&lt;tt&gt;</code></td>
<td>Teletype</td>
</tr>
<tr>
<td><code>&lt;em&gt;</code></td>
<td>Emphasized</td>
</tr>
<tr>
<td><code>&lt;strong&gt;</code></td>
<td>Strong</td>
</tr>
<tr>
<td><code>&lt;del&gt;</code></td>
<td>Deleted</td>
</tr>
<tr>
<td><code>&lt;ins&gt;</code></td>
<td>Inserted</td>
</tr>
<tr>
<td><code>&lt;pre&gt;</code></td>
<td>Preformatted</td>
</tr>
<tr>
<td><code>&lt;code&gt;</code></td>
<td>Source code text</td>
</tr>
<tr>
<td><code>&lt;samp&gt;</code></td>
<td>Sample code text</td>
</tr>
<tr>
<td><code>&lt;kbd&gt;</code></td>
<td>Keyboard key</td>
</tr>
<tr>
<td><code>&lt;var&gt;</code></td>
<td>Variable name</td>
</tr>
<tr>
<td><code>&lt;cite&gt;</code></td>
<td>Citation</td>
</tr>
<tr>
<td><code>&lt;blockquote&gt;</code></td>
<td>Quotation</td>
</tr>
<tr>
<td><code>&lt;q&gt;</code></td>
<td>Short quotation</td>
</tr>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Address block</td>
</tr>
<tr>
<td><code>&lt;abbr&gt;</code></td>
<td>Abbreviation</td>
</tr>
<tr>
<td><code>&lt;acronym&gt;</code></td>
<td>Acronym</td>
</tr>
<tr>
<td><code>&lt;dfn&gt;</code></td>
<td>Definition</td>
</tr>
<tr>
<td><code>&lt;font&gt;</code></td>
<td>Font †</td>
</tr>
<tr>
<td><code>&lt;bdo&gt;</code></td>
<td>Bidirectional override</td>
</tr>
<tr>
<td><code>&lt;xmp&gt;</code></td>
<td>XMP</td>
</tr>
<tr>
<td>other tags</td>
<td>Attributes common to almost all other tags</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;i&gt;</code></td>
<td>Italic</td>
</tr>
<tr>
<td><code>&lt;b&gt;</code></td>
<td>Bold</td>
</tr>
<tr>
<td><code>&lt;s&gt;</code></td>
<td>Strike-through text †</td>
</tr>
<tr>
<td><code>&lt;u&gt;</code></td>
<td>Underlined text †</td>
</tr>
<tr>
<td><code>&lt;big&gt;</code></td>
<td>Bigger</td>
</tr>
<tr>
<td><code>&lt;small&gt;</code></td>
<td>Smaller</td>
</tr>
<tr>
<td><code>&lt;sub&gt;</code></td>
<td>Subscript</td>
</tr>
<tr>
<td><code>&lt;sup&gt;</code></td>
<td>Superscript</td>
</tr>
<tr>
<td><code>&lt;tt&gt;</code></td>
<td>Teletype</td>
</tr>
<tr>
<td><code>&lt;em&gt;</code></td>
<td>Emphasized</td>
</tr>
<tr>
<td><code>&lt;strong&gt;</code></td>
<td>Strong</td>
</tr>
<tr>
<td><code>&lt;del&gt;</code></td>
<td>Deleted</td>
</tr>
<tr>
<td><code>&lt;ins&gt;</code></td>
<td>Inserted</td>
</tr>
<tr>
<td><code>&lt;pre&gt;</code></td>
<td>Preformatted</td>
</tr>
<tr>
<td><code>&lt;code&gt;</code></td>
<td>Source code text</td>
</tr>
<tr>
<td><code>&lt;samp&gt;</code></td>
<td>Sample code text</td>
</tr>
<tr>
<td><code>&lt;kbd&gt;</code></td>
<td>Keyboard key</td>
</tr>
<tr>
<td><code>&lt;var&gt;</code></td>
<td>Variable name</td>
</tr>
<tr>
<td><code>&lt;cite&gt;</code></td>
<td>Citation</td>
</tr>
<tr>
<td><code>&lt;blockquote&gt;</code></td>
<td>Quotation</td>
</tr>
<tr>
<td><code>&lt;q&gt;</code></td>
<td>Short quotation</td>
</tr>
<tr>
<td><code>&lt;address&gt;</code></td>
<td>Address block</td>
</tr>
<tr>
<td><code>&lt;abbr&gt;</code></td>
<td>Abbreviation</td>
</tr>
<tr>
<td><code>&lt;acronym&gt;</code></td>
<td>Acronym</td>
</tr>
<tr>
<td><code>&lt;dfn&gt;</code></td>
<td>Definition</td>
</tr>
<tr>
<td><code>&lt;font&gt;</code></td>
<td>Font †</td>
</tr>
<tr>
<td><code>&lt;bdo&gt;</code></td>
<td>Bidirectional override</td>
</tr>
<tr>
<td><code>&lt;xmp&gt;</code></td>
<td>XMP</td>
</tr>
<tr>
<td>other tags</td>
<td>Attributes common to almost all other tags</td>
</tr>
</tbody>
</table>

† = deprecated
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;img&gt;</code></td>
<td>*align=top</td>
<td>bottom</td>
</tr>
<tr>
<td></td>
<td>alt=alternatetext</td>
<td>Description of the image for text-only browsers</td>
</tr>
<tr>
<td></td>
<td>border=npixels</td>
<td>Border width around the image †</td>
</tr>
<tr>
<td></td>
<td>height=npixels</td>
<td>percent%</td>
</tr>
<tr>
<td></td>
<td>hspace=npixels</td>
<td>Blank space on the left and right side of image †</td>
</tr>
<tr>
<td></td>
<td>ismap=url</td>
<td>URL for server-side image map</td>
</tr>
<tr>
<td></td>
<td>longdesc=url</td>
<td>URL containing a long description of the image</td>
</tr>
<tr>
<td></td>
<td>src=url</td>
<td>URL of the image</td>
</tr>
<tr>
<td></td>
<td>usemap=url</td>
<td>URL for client-side image map</td>
</tr>
<tr>
<td></td>
<td>vspace=npixels</td>
<td>Blank space on top and bottom of image †</td>
</tr>
<tr>
<td></td>
<td>width=npixels</td>
<td>percent%</td>
</tr>
<tr>
<td><code>&lt;map&gt;</code></td>
<td>id=id</td>
<td>Unique ID for the map tag</td>
</tr>
<tr>
<td></td>
<td>name=name</td>
<td>Unique name for the map tag</td>
</tr>
<tr>
<td><code>&lt;area&gt;</code></td>
<td>alt=alternatetext</td>
<td>Description of area for text-only browsers</td>
</tr>
<tr>
<td></td>
<td>coords=left,top,right,bottom</td>
<td>cx,cy,radius</td>
</tr>
<tr>
<td></td>
<td>href=url</td>
<td>Target URL of area</td>
</tr>
<tr>
<td></td>
<td>nohref=true</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>shape=rectangle</td>
<td>circle</td>
</tr>
<tr>
<td></td>
<td>target=blank</td>
<td>_parent</td>
</tr>
</tbody>
</table>

† = deprecated
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| **<table>** Table | **align**=left|center|right  
**bgcolor**=rgb(r,g,b)|#rrggbb  
**border**=npixels  
**cellpadding**=npixels|percent%  
**cellspacing**=npixels|percent%  
**frame**=void|above|below|lhs|rhs|hsides|vsides|box|border  
**rules**=none|groups|rows|cols|all  
**summary**=summary  
**width**=npixels|percent% |
| **Table alignment †** | **Table background color †** |
| **<tr>** Table row | **align**=left|center|right|justify|char  
**bgcolor**=rgb(r,g,b)|#rrggbb  
**char**=character  
**charoff**=npixels|percent%  
**valign**=top|middle|bottom|baseline |
| **Horizontal text alignment** | **Row background color †** |
| **<td>** Table cell | **abbr**=content  
**align**=left|center|right|justify|char  
**axis**=category  
**bgcolor**=rgb(r,g,b)|#rrggbb|color  
**char**=character  
**charoff**=npixels|percent%  
**colspan**=ncolumns  
**headers**=headerid  
**height**=npixels  
**nowrap**  
**rowspan**=nrows  
**scope**=col|colgroup|row|rowgroup  
**valign**=top|middle|bottom|baseline  
**width**=npixels|percent% |
| **Abbreviated content in a cell** | **Horizontal text alignment** |
| **Cell name** | **Cell background color †** |
| **Character to align text on, if align=char** | **Alignment offset to first character, if align=char** |
| **Number of columns this cell spans on** | **Cell header information for text-only browsers** |
| **Cell height †** | **Text in cell stays on a single line †** |
| **Number of rows this cell spans on** | **Target for cell header information** |
| **Vertical text alignment** | **Cell width †** |

† = deprecated
### 7-bit ASCII Table

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Hex</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>NUL</td>
<td>32</td>
<td>20</td>
<td>space</td>
<td>64</td>
<td>40</td>
<td>@</td>
<td>96</td>
<td>60</td>
<td>`</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>SOH</td>
<td>33</td>
<td>21</td>
<td>!</td>
<td>65</td>
<td>41</td>
<td>A</td>
<td>97</td>
<td>61</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>STX</td>
<td>34</td>
<td>22</td>
<td>&quot;</td>
<td>66</td>
<td>42</td>
<td>B</td>
<td>98</td>
<td>62</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>ETX</td>
<td>35</td>
<td>23</td>
<td>#</td>
<td>67</td>
<td>43</td>
<td>C</td>
<td>99</td>
<td>63</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>EOT</td>
<td>36</td>
<td>24</td>
<td>$</td>
<td>68</td>
<td>44</td>
<td>D</td>
<td>100</td>
<td>64</td>
<td>d</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>ENQ</td>
<td>37</td>
<td>25</td>
<td>%</td>
<td>69</td>
<td>45</td>
<td>E</td>
<td>101</td>
<td>65</td>
<td>e</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>ACK</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
<td>70</td>
<td>46</td>
<td>F</td>
<td>102</td>
<td>66</td>
<td>f</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>BEL</td>
<td>39</td>
<td>27</td>
<td>'</td>
<td>71</td>
<td>47</td>
<td>G</td>
<td>103</td>
<td>67</td>
<td>g</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>BS</td>
<td>40</td>
<td>28</td>
<td>(</td>
<td>72</td>
<td>48</td>
<td>H</td>
<td>104</td>
<td>68</td>
<td>h</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>TAB</td>
<td>41</td>
<td>29</td>
<td>)</td>
<td>73</td>
<td>49</td>
<td>I</td>
<td>105</td>
<td>69</td>
<td>i</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>LF</td>
<td>42</td>
<td>2A</td>
<td>*</td>
<td>74</td>
<td>4A</td>
<td>J</td>
<td>106</td>
<td>6A</td>
<td>j</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>VT</td>
<td>43</td>
<td>2B</td>
<td>+</td>
<td>75</td>
<td>4B</td>
<td>K</td>
<td>107</td>
<td>6B</td>
<td>k</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>FF</td>
<td>44</td>
<td>2C</td>
<td>,</td>
<td>76</td>
<td>4C</td>
<td>L</td>
<td>108</td>
<td>6C</td>
<td>l</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>CR</td>
<td>45</td>
<td>2D</td>
<td>-</td>
<td>77</td>
<td>4D</td>
<td>M</td>
<td>109</td>
<td>6D</td>
<td>m</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>SO</td>
<td>46</td>
<td>2E</td>
<td>.</td>
<td>78</td>
<td>4E</td>
<td>N</td>
<td>110</td>
<td>6E</td>
<td>n</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>SI</td>
<td>47</td>
<td>2F</td>
<td>/</td>
<td>79</td>
<td>4F</td>
<td>O</td>
<td>111</td>
<td>6F</td>
<td>o</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>DLE</td>
<td>48</td>
<td>30</td>
<td>0</td>
<td>80</td>
<td>50</td>
<td>P</td>
<td>112</td>
<td>70</td>
<td>p</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>DC1</td>
<td>49</td>
<td>31</td>
<td>1</td>
<td>81</td>
<td>51</td>
<td>Q</td>
<td>113</td>
<td>71</td>
<td>q</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>DC2</td>
<td>50</td>
<td>32</td>
<td>2</td>
<td>82</td>
<td>52</td>
<td>R</td>
<td>114</td>
<td>72</td>
<td>r</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>DC3</td>
<td>51</td>
<td>33</td>
<td>3</td>
<td>83</td>
<td>53</td>
<td>S</td>
<td>115</td>
<td>73</td>
<td>s</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>DC4</td>
<td>52</td>
<td>34</td>
<td>4</td>
<td>84</td>
<td>54</td>
<td>T</td>
<td>116</td>
<td>74</td>
<td>t</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>NAK</td>
<td>53</td>
<td>35</td>
<td>5</td>
<td>85</td>
<td>55</td>
<td>U</td>
<td>117</td>
<td>75</td>
<td>u</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>SYN</td>
<td>54</td>
<td>36</td>
<td>6</td>
<td>86</td>
<td>56</td>
<td>V</td>
<td>118</td>
<td>76</td>
<td>v</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>ETB</td>
<td>55</td>
<td>37</td>
<td>7</td>
<td>87</td>
<td>57</td>
<td>W</td>
<td>119</td>
<td>77</td>
<td>w</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>CAN</td>
<td>56</td>
<td>38</td>
<td>8</td>
<td>88</td>
<td>58</td>
<td>X</td>
<td>120</td>
<td>78</td>
<td>x</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>EM</td>
<td>57</td>
<td>39</td>
<td>9</td>
<td>89</td>
<td>59</td>
<td>Y</td>
<td>121</td>
<td>79</td>
<td>y</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>SUB</td>
<td>58</td>
<td>3A</td>
<td>:</td>
<td>90</td>
<td>5A</td>
<td>Z</td>
<td>122</td>
<td>7A</td>
<td>z</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>ESC</td>
<td>59</td>
<td>3B</td>
<td>;</td>
<td>91</td>
<td>5B</td>
<td>]</td>
<td>123</td>
<td>7B</td>
<td>]</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>FS</td>
<td>60</td>
<td>3C</td>
<td>&lt;</td>
<td>92</td>
<td>5C</td>
<td>\</td>
<td>124</td>
<td>7C</td>
<td>\</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>GS</td>
<td>61</td>
<td>3D</td>
<td>=</td>
<td>93</td>
<td>5D</td>
<td>{</td>
<td>125</td>
<td>7D</td>
<td>{</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>RS</td>
<td>62</td>
<td>3E</td>
<td>&gt;</td>
<td>94</td>
<td>5E</td>
<td>^</td>
<td>126</td>
<td>7E</td>
<td>~</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>US</td>
<td>63</td>
<td>3F</td>
<td>?</td>
<td>95</td>
<td>5F</td>
<td>_</td>
<td>127</td>
<td>7F</td>
<td>DEL Delete</td>
</tr>
</tbody>
</table>

Characters 0-31 and 127 are non-printable.

---

**ASCII Table**

Display an ASCII table

```
ascii
man ascii
```

Prompt for pressing a key and display its ASCII value in decimal, octal, and hex

```
showkey -a
```
Confidentiality, Integrity, and Availability (aka the CIA triad) are the basic policies of Information Security. Confidentiality ensures that access to information is limited to those with the correct rights, integrity ensures that information has not been improperly modified, and availability ensures that a system is operable and functioning.

A threat is an entity, circumstance, or event with the potential to adversely impact a computer system through unauthorized access, destruction, disclosure, modification of data, or Denial of Service.
A vulnerability is a weakness in a computer system that could be exploited by a threat source.
A risk is the probability that a particular security threat will exploit a system vulnerability, according to the risk equation: risk = threat × vulnerability.
An attack is the attempt to gain unauthorized access to a computer system’s services, resources, or information, or the attempt to compromise a computer system’s confidentiality, integrity, or availability.

### Vulnerability management life cycle

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation of a baseline</td>
<td>Identify and prioritize critical assets</td>
</tr>
<tr>
<td>Vulnerability assessment</td>
<td>Identify and prioritize vulnerabilities. Examine the abilities of a system,</td>
</tr>
<tr>
<td></td>
<td>applications, security procedures, and controls, to withstand an attack</td>
</tr>
<tr>
<td>Risk assessment</td>
<td>Analyze and evaluate risks to determine possible incidents, their likelihood,</td>
</tr>
<tr>
<td></td>
<td>their consequences, and the tolerance of each critical asset for such events.</td>
</tr>
<tr>
<td>Remediation</td>
<td>Reduce the severity of vulnerabilities (via action plans, patches, hotfixes,</td>
</tr>
<tr>
<td></td>
<td>etc.)</td>
</tr>
<tr>
<td>Verification</td>
<td>Verify all the previous phases (via scanners, reports, etc.)</td>
</tr>
<tr>
<td>Monitor</td>
<td>Monitor regularly the system to maintain the required level of security</td>
</tr>
</tbody>
</table>

The Common Vulnerability Scoring System (CVSS) assessment consists of three metrics for measuring vulnerabilities: base metrics, temporal metrics, and environmental metrics. Each metric sets a severity score from 1 to 10.

Common Vulnerabilities and Exposures (CVE) is a public list of identifiers for software vulnerabilities.

### Access control types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discretionary Access Control</td>
<td>Allows the user that has access to the resource to decide with whom to share</td>
</tr>
<tr>
<td>(DAC) aka need-to-know</td>
<td>it. File access is regulated by user and group permissions. In Linux, this is</td>
</tr>
<tr>
<td></td>
<td>the standard access model.</td>
</tr>
<tr>
<td>Mandatory Access Control</td>
<td>A particular user can access a resource only if they have been given explicit</td>
</tr>
<tr>
<td>(MAC)</td>
<td>access right to it. The end user is not allowed to choose who can access the</td>
</tr>
<tr>
<td></td>
<td>resource, or to pass privileges. In Linux, this is implemented by SELinux.</td>
</tr>
<tr>
<td>Role-Based Access Control</td>
<td>Access permissions are based on the access policies determined by the system.</td>
</tr>
<tr>
<td>(RBAC)</td>
<td>Users are assigned access to resources on a one-to-one basis.</td>
</tr>
<tr>
<td>Rule-Based Access Control</td>
<td>Endpoint devices (e.g. firewalls) verify the requests to access network</td>
</tr>
<tr>
<td>(RuBAC)</td>
<td>resources against a set of rules based on IP addresses, port numbers, etc.</td>
</tr>
</tbody>
</table>
The Metasploit Framework is a well-known tool and the de facto standard for developing and exploiting security vulnerabilities in systems and applications. The GUI equivalent is Armitage.

Metasploit has a modular architecture. **Exploit** modules are the basic modules used to encapsulate an exploit. **Payload** modules can be **singles** (self-contained), **stagers** (modules that establish a network connection to the attacked system), and **stages** (downloaded by stagers). **Auxiliary** modules are used to perform one-time actions such as port scanning, DoS, or fuzzing. **NOP** modules generate no-op instructions (0x90 for x86 microchips) used to keep the payload size consistent by padding out buffers.

The steps for carrying out an attack are, in order: configuring an exploit, setting up the exploit options, selecting a target, selecting a payload, and launching the exploit.

```
msfconsole  Launch the Metasploit Framework CLI
msfpupdate  Update the Metasploit Framework
msfvenom    Generate and encode a payload for an exploit. Replaces the old msfpayload and msfencode tools
```

### Base commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Show the list of Metasploit commands</td>
</tr>
<tr>
<td>help command</td>
<td>Show detailed help about command</td>
</tr>
<tr>
<td>db_status</td>
<td>Check database connection status</td>
</tr>
<tr>
<td>creds</td>
<td>Display all credentials in the database</td>
</tr>
<tr>
<td>use module</td>
<td>Load and use module</td>
</tr>
<tr>
<td>setg variable value</td>
<td>Assign value to a global variable</td>
</tr>
<tr>
<td>getg variable</td>
<td>Get the value of a global variable</td>
</tr>
<tr>
<td>unsetg variable</td>
<td>Unset the value of a global variable</td>
</tr>
<tr>
<td>connect host port</td>
<td>Connect to a remote host on port</td>
</tr>
<tr>
<td>sessions</td>
<td>Display information about active sessions</td>
</tr>
<tr>
<td>threads</td>
<td>Display information about background threads and manipulate them</td>
</tr>
<tr>
<td>banner</td>
<td>Display a Metasploit banner</td>
</tr>
<tr>
<td>history</td>
<td>Show Metasploit command history</td>
</tr>
</tbody>
</table>

### Module commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>info</td>
<td>See information about the currently loaded module</td>
</tr>
<tr>
<td>show payloads</td>
<td>Show the list of compatible payloads for the currently loaded module</td>
</tr>
<tr>
<td>show options</td>
<td>Show all options (variables) available for the module, along with their descriptions and set values</td>
</tr>
<tr>
<td>set variable value</td>
<td>Assign value to a context-specific variable</td>
</tr>
<tr>
<td>get variable</td>
<td>Get the value of a context-specific variable</td>
</tr>
<tr>
<td>unset variable</td>
<td>Unset the value of a context-specific variable</td>
</tr>
<tr>
<td>check host</td>
<td>Check if host is vulnerable</td>
</tr>
<tr>
<td>reload</td>
<td>Reload the module</td>
</tr>
<tr>
<td>rexploit rerun</td>
<td>Reload and execute the module</td>
</tr>
<tr>
<td>exploit run</td>
<td>Execute the module</td>
</tr>
</tbody>
</table>
Meterpreter is an advanced and dynamically extensible payload for a Metasploit attack that provides the attacker with an interactive shell (Meterpreter session) to the target remote machine. This session is obtained by running from Metasploit an exploit with the appropriate payload e.g. `php/meterpreter/reverse_tcp`. The payload is deployed using in-memory DLL injection.

### Meterpreter commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Show a list of all Meterpreter commands</td>
</tr>
<tr>
<td>background</td>
<td>Send the Meterpreter session to background and return to the MSF CLI</td>
</tr>
<tr>
<td>cat file</td>
<td>Display the contents of <code>file</code></td>
</tr>
<tr>
<td>edit file</td>
<td>Edit a text <code>file</code> (via Vim)</td>
</tr>
<tr>
<td>ls</td>
<td>List files on the target machine</td>
</tr>
<tr>
<td>ps</td>
<td>Display processes on the target machine</td>
</tr>
<tr>
<td>download file</td>
<td>Download <code>file</code> from the target machine</td>
</tr>
<tr>
<td>upload file path</td>
<td>Upload <code>file</code> from the local machine to the <code>path</code> on the target machine</td>
</tr>
<tr>
<td>execute -f command</td>
<td>Execute a <code>command</code> on the target machine</td>
</tr>
<tr>
<td>resource file</td>
<td>Execute on the target machine the Meterpreter commands listed in the local <code>file</code></td>
</tr>
<tr>
<td>shell</td>
<td>Run an OS shell on the target machine</td>
</tr>
<tr>
<td>sysinfo</td>
<td>Get information about the target machine: OS, machine name, etc.</td>
</tr>
<tr>
<td>ipconfig</td>
<td>Display network configuration of the target machine</td>
</tr>
<tr>
<td>localtime</td>
<td>Display date and time of the target machine</td>
</tr>
<tr>
<td>clearev</td>
<td>On a MS Windows target machine, clear all Application, Security, and System logs</td>
</tr>
<tr>
<td>webcam_list</td>
<td>List all operative webcams on the target machine</td>
</tr>
<tr>
<td>webcam_snap</td>
<td>Take a snapshot from the webcam on the target machine</td>
</tr>
<tr>
<td>quit</td>
<td>End the Meterpreter session</td>
</tr>
<tr>
<td>exit</td>
<td>End the Meterpreter session</td>
</tr>
</tbody>
</table>
Aircrack-ng

**aircrack-ng options file**

Crack WEP or WPA/WPA2-PSK keys from the capture file (in .cap or .ivs format).

Possible options:

- `-a n` Attack mode (n=1 for WEP, n=2 for WPA/WPA2-PSK)
- `-e essid` Specify the Access Point to use
- `-K` Use the Korek WEP cracking technique
- `-z` Use the PTW WEP cracking technique
- `-k n` Disable the Korek WEP attack number n (where n is between 1 and 17)
- `-n len` Specify WEP key length
- `-s` Show WEP key in ASCII while cracking
- `-w file` Wordlist file to use for WEP or WPA/WPA2 key cracking

**aireplay-ng attack options**

Replay packets to perform an attack, where attack is one of:

- `-0` Deauthentication attack
- `-1` Fake authentication attack
- `-2` Interactive packet replay attack
- `-3` ARP Request replay attack
- `-4` Chopchop attack
- `-5` Fragmentation attack (against WEP)
- `-6` Caffe Latte attack
- `-7` Hirte attack
- `-8` WPA Migration Mode attack
- `-9` Injection test

**airodump options interface**

Capture packets by listening to the network interface.

Possible options:

- `--ivs` Save only captured IVs
- `-w file` Write sniffed packets in a capture file
- `-o format` Use format for the capture file: may be cap, ivs, csv, gps, kismet, netxml, or logcsv
How to crack WEP

1. `airmon-ng start wlan0` - Start the wireless NIC into monitor mode on the same channel as the AP, and test injection capabilities to the AP.

2. `airodump-ng --iw -w capture wlan0` - Discover the list of active wireless machines. Note ESSID and BSSID of the target AP (let us assume they are respectively `ap_essid` and `ap_mac`).

3. `aireplay-ng -l 0 -e ap_essid \ -a ap_mac -h mac wlan0` - Do a fake authentication with the target AP using your NIC (with MAC address `mac`).

4. `aireplay-ng -3 -b ap_mac -h mac wlan0` - To capture a large number of IVs in a short time, run `aireplay-ng` in ARP Request replay mode.

5. `aircrack-ng -s capture.ivs` - Once `airodump-ng` has captured at least 50000 IVs, crack the WEP key.

How to crack WPA-PSK or WPA2-PSK

1. `airmon-ng start wlan0` - Put the wireless NIC into monitor mode.

2. `airodump-ng -w capture wlan0` - Discover the list of active wireless machines. Note BSSID of the target client (let us assume it's `cl_mac`).

3. `aireplay-ng --deauth 11 -a cl_mac` - Deauthenticate the client. The client will try to authenticate again, and `airodump-ng` will capture the authentication packet sent during the WPA/WPA2 four-way handshake.

4. `aircrack-ng -a 2 -w capture.cap` - Analyze the capture dump to crack the WPA/WPA2 key.

The only way to crack WPA/WPA2 is to sniff the Pairwise Master Key associated with the four-way handshake authentication process. Therefore it is important to choose a complex WPA/WPA2 random password at least 20 characters long.

How to decloak a hidden SSID

1. `airmon-ng start wlan0` - Put the wireless NIC into monitor mode.

2. `airodump-ng --iw -w capture wlan0` - Discover the list of active wireless clients. Find the one where the ESSID is hidden (it shows only its string length) and note its BSSID (let us assume it's `cl_mac`).

3. `aireplay-ng --deauth 11 -a cl_mac` - Deauthenticate the client.

4. The output of `airodump-ng` will now show the hidden ESSID.

How to perform a MitM attack

1. `airmon-ng start wlan0` - Put the wireless NIC into monitor mode.

2. `airodump-ng --iw -w capture wlan0` - Discover the list of active wireless clients. Note BSSID of the target client (let us assume it's `cl_mac`). Note ESSID and BSSID of the AP you are currently accessing (let us assume they are respectively `ap_essid` and `ap_mac`).

3. `aireplay-ng --deauth 11 -a cl_mac` - Deauthenticate the client.

4. `aireplay-ng -l 0 -e ap_essid \ -a ap_mac -h cl_mac wlan0` - Perform a fake authentication attack, associating your NIC with the AP you are currently accessing.
The Firmware Mod Kit is a toolkit to extract, deconstruct, modify, rebuild, and flash firmware images for Linux-based routers, IoT devices, embedded devices, and most other devices that use common firmware formats and filesystems such as TRX/uImage and SquashFS/CramFS.

It can be used by an attacker after compromising a device, to maintain access.

- extract-firmware.sh: Extract the firmware
- build-firmware.sh: Rebuild the firmware
- ddwrt-gui-extract.sh: Extract Web GUI files from extracted DD-WRT firmware
- ddwrt-gui-rebuild.sh: Restore modified Web GUI files to extracted DD-WRT firmware
### Security tools - network

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arpspoof</td>
<td>ARP spoofing tool</td>
</tr>
<tr>
<td>arpoison</td>
<td>ARP cache update utility. Can be used to craft custom ARP packets</td>
</tr>
<tr>
<td>arpsraw</td>
<td>ARP spoofing detection tool</td>
</tr>
<tr>
<td>arpon</td>
<td>ARP handler inspector. Useful to detect ARP spoofing</td>
</tr>
<tr>
<td>arpwatch</td>
<td>Tracker of MAC to IP address pairings. Useful to detect ARP spoofing</td>
</tr>
<tr>
<td>ettercap</td>
<td>Network security tool for ARP poisoning and man-in-the-middle attacks over the LAN</td>
</tr>
<tr>
<td>macchanger</td>
<td>Tool to perform MAC spoofing. Changes the NIC to a random MAC address</td>
</tr>
<tr>
<td>macof</td>
<td>DoS tool for MAC flooding</td>
</tr>
<tr>
<td>yersinia</td>
<td>DoS tool for DHCP starvation attack</td>
</tr>
<tr>
<td>dhcpstarv</td>
<td>DoS tool for DHCP starvation attack</td>
</tr>
<tr>
<td>dhcpig</td>
<td>DoS tool for DHCP starvation attack</td>
</tr>
<tr>
<td>nbtscan</td>
<td>Network scanner for NetBIOS name information</td>
</tr>
<tr>
<td>p0f</td>
<td>Passive traffic fingerprinting tool. Identifies hosts performing any incidental TCP/IP communication</td>
</tr>
<tr>
<td>bannergrab</td>
<td>Network service banner grabbing tool. Sends a trigger to the service and collects basic information</td>
</tr>
<tr>
<td>nscan</td>
<td>Fast network scanner optimized for Internet-wide scanning</td>
</tr>
<tr>
<td>zmap</td>
<td>Fast single packet network scanner. An improved version of nmap designed for Internet-wide scanning</td>
</tr>
<tr>
<td>masscan</td>
<td>Fast Internet port scanner</td>
</tr>
<tr>
<td>fragrouter</td>
<td>IDS evasion toolkit. Reroutes network traffic</td>
</tr>
<tr>
<td>dnsspoof</td>
<td>DNS spoofing tool. Forges replies to DNS queries on the LAN</td>
</tr>
<tr>
<td>responder</td>
<td>LLNR, NBT-NS, and MDNS poisoner</td>
</tr>
<tr>
<td>scapy</td>
<td>Packet manipulation tool. Features packet forging, decoding, injection, and other network operations</td>
</tr>
<tr>
<td>mitmf</td>
<td>Framework for MitM attacks</td>
</tr>
<tr>
<td>loki</td>
<td>Firewall evasion tool that encapsulates commands into the payload of ICMP packets</td>
</tr>
<tr>
<td>hts</td>
<td>HTTPTunnel server. Used in conjunction with the HTTPTunnel client htc to tunnel network connections through pure HTTP traffic (GET and POST requests), hence bypassing restrictive firewalls or proxies</td>
</tr>
<tr>
<td>htc</td>
<td>HTTPTunnel client</td>
</tr>
<tr>
<td>iodined</td>
<td>Firewall evasion tool. Tunnels IPv4 traffic through a DNS server. Replaces the obsolete tcp-over-dns</td>
</tr>
<tr>
<td>iodine</td>
<td>Client for iodined</td>
</tr>
<tr>
<td>loic</td>
<td>Low Orbit Ion Cannon, a GUI tool for network stress testing and DoS/DDoS attacks</td>
</tr>
<tr>
<td>hoic</td>
<td>High Orbit Ion Cannon, a GUI tool for network stress testing and DoS/DDoS attacks</td>
</tr>
<tr>
<td>zz</td>
<td>Zombie Zapper, a countermeasure tool capable of stopping DDoS packet flooding attacks carried out by Trin00, TFN, Stacheldraht, etc.</td>
</tr>
</tbody>
</table>
Security tools - Wi-Fi

- **aircrack-ng**: WEP and WPA/WPA2-PSK key cracker. Part of the Aircrack-ng suite for Wi-Fi network cracking.
- **airodump-ng**: Packet sniffer.
- **airoscript-ng**: User-friendly interface for aircrack-ng.
- **airodump-ng**: Tool to generate graphs of relationships between wireless devices, using data from airodump-ng.
- **aireplay-ng**: Packet injector.
- **packetforge-ng**: Generator of encrypted packets for injection.
- **airbase-ng**: Tool for attacks against wireless clients (and not Access Points).
- **airserv-ng**: Tool to access the wireless NIC from other machines.
- **tkiptun-ng**: WPA-TKIP attack tool.
- **wesside-ng**: WEP key recovery tool.
- **airdecloak-ng**: Tool to remove WEP cloaking from PCAP dump files.
- **airolib-ng**: Manager for ESSID and password lists. These are used for WPA and WPA2 cracking.
- **airdecap-ng**: Decryption tool for WEP and WPA.
- **airmon-ng**: Tool to set up wireless NICs in monitor mode.
- **airtun-ng**: Virtual tunnel interface creator.
- **airdriver-ng**: Tool that displays information about wireless drivers on the system.
- **airdrop-ng**: Tool to force deauthentication of users.
- **ivstools**: Tool to extract or merge IVs from a PCAP dump file.
- **kstats**: Program that displays statistical FMS algorithm votes for an IVs dump with a specific WEP key.
- **makeivs**: Tool to generate a dummy IVs dump file with a specific WEP key.
- **versuck-ng**: Tool to calculate the default WEP key for Verizon Actiontec wireless routers.
- **easside-ng**: Tool to communicate with an Access Point without knowing its WEP key.
- **buddy-ng**: Helper server for easside-ng.
- **fern-wifi-cracker**: Wi-Fi auditing and attack tool, with GUI.
- **airsnort**: WEP key cracker using the Fluhrer, Mantin, and Shamir attack (FMS).
- **wepattack**: WEP key cracker. Takes a dumpfile as input.
- **WEPCrack**: WEP key cracker.
- **airfart**: GUI tool that detects Wi-Fi devices and displays their signal strength.
- **cowpatty**: WPA-PSK key cracker via dictionary attack. Part of the coWPAtty package.
- **genpmk**: WPA-PSK key cracker via precomputation attack.
kismet

Wireless sniffer and IDS

hackrf_info

Tool that probes the HackRF One device and shows its configuration. HackRF One is an open source SDR (Software Defined Radio) peripheral for testing RF systems. It is able to transmit and receive radio signals in the 1 MHz - 6 GHz range and can be used to carry out attacks such as replay, jamming, fuzzing, BlueBorne, etc. Usually used with the Ubuntu, Gentoo, and Pentoo distros

rfcat

ISM band radio multipurpose tool

rfcat-rolljam.py

Exploit for the rolling code attack. Jams, captures, and replays radio signals

scapy-radio

Modified version of the scapy packet manipulation tool to include RF capabilities

RFCrack.py

Radio Frequency communications scanner and test workbench. Can be used to perform the rolling code attack

btscanner

Bluetooth scanner with ncurses UI

bluescan

Bluetooth scanner

bluediving

Bluetooth pentesting suite

bluesniff

Bluetooth device discovery (wardriving) utility. Proof of concept
<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlmap</td>
<td>Automatic SQL injection exploitation/pentesting tool. The GUI version is Tyrant SQL</td>
</tr>
<tr>
<td>mole</td>
<td>Automatic SQL injection exploitation/pentesting tool</td>
</tr>
<tr>
<td>jsql</td>
<td>jSQL Injection, a GUI tool to retrieve database information from a remote server</td>
</tr>
<tr>
<td>sqlsus</td>
<td>Fast and efficient SQL injection and takeover tool for MySQL</td>
</tr>
<tr>
<td>ISR-sqlget</td>
<td>Blind SQL injection tool</td>
</tr>
<tr>
<td>fatrmat</td>
<td>TheFatRat, a Remote Access Trojan and exploiting tool</td>
</tr>
<tr>
<td>horsepill</td>
<td>Ramdisk-based rootkit</td>
</tr>
<tr>
<td>amap</td>
<td>Application Mapper, a scanning and fingerprinting tool for pentesting</td>
</tr>
<tr>
<td>amapcrap</td>
<td>Fuzzer that sends random data to a TCP/UDP port and gathers a response for use by amap</td>
</tr>
<tr>
<td>svmap</td>
<td>Scanner for discovering VoIP SIP devices. Part of the SIPVicious package</td>
</tr>
<tr>
<td>svwar</td>
<td>Scanner for identifying active extensions on a PBX</td>
</tr>
<tr>
<td>svcrack</td>
<td>Online password cracker for SIP PBX</td>
</tr>
<tr>
<td>svcrash</td>
<td>Countermeasure for unauthorized svwar and svcrack scans</td>
</tr>
<tr>
<td>svreport</td>
<td>SIP audit session manager and reports exporter</td>
</tr>
<tr>
<td>ike-scan</td>
<td>Tool for discovering IKE hosts (IPsec VPN servers) and determine which IKE implementation they use</td>
</tr>
<tr>
<td>ddosim</td>
<td>Layer 7 DDoS simulator. Simulates zombie hosts connecting to a specific application on the target host</td>
</tr>
<tr>
<td>fat</td>
<td>Firmware Analysis Toolkit, a toolkit to find and analyze vulnerabilities in the firmware of IoT devices and embedded devices</td>
</tr>
<tr>
<td>balbuzard</td>
<td>Tool to extract patterns of interest (e.g. URLs, IP addresses, typical malware strings) from malicious files</td>
</tr>
<tr>
<td>bbcrack</td>
<td>Tool to bruteforce typical malware obfuscation transforms (XOR, ROL, ADD, etc.) and discover the algorithms and keys used. Based on patterns of interest</td>
</tr>
<tr>
<td>bbharvest</td>
<td>Tool to extract all patterns of interest found when applying typical malware obfuscation transforms (XOR, ROL, ADD, etc.) trying all possible keys</td>
</tr>
<tr>
<td>bbtrans</td>
<td>Tool to apply any of the malware obfuscation transforms from bbcrack to a file</td>
</tr>
</tbody>
</table>
w3af
- Web Application Attack and Audit Framework, GUI. Finds and exploits web application vulnerabilities

w3af-console
- Web Application Attack and Audit Framework, CLI version

nikto
- Webserver vulnerability scanner

skipfish
- Webserver vulnerability scanner

uniscan
- Simple webserver vulnerability scanner

burp
- Burp Suite, a comprehensive testing platform for web applications security. Allows intercepting and modifying HTTP/HTTPS requests and replies to perform HTTP session hijacking

zap
- OWASP Zed Attack Proxy, a pentesting GUI tool for web applications

webscarab
- Tool for testing the security of web applications. Runs as a proxy

arachni_web
- Modular framework and penetration testing platform to test the security of web applications

nuclei
- Penetration tool, based on templates, for configurable targeted scanning

xsser
- Automatic framework to detect, exploit, and report XSS vulnerabilities in web applications

vega
- GUI tool to test the security of web applications

paros
- GUI tool with HTTP/HTTPS proxy functionality for assessing web application vulnerabilities

wapiti
- Vulnerability scanner for web applications

httpbee
- Web application testing tool

wsfuzzer
- Fuzzer for black box testing of web services (HTTP and SOAP)

ffuf
- Fast web fuzzer

httprint
- Web server fingerprinting tool

wafw00f
- Tool that detects and fingerprints a WAF (Web App Firewall) i.e. a firewall that protects a webserver

wafninja
- Tool that circumvents a WAF by automating the steps necessary for bypassing input validation

whatweb
- Web scanner. Detects and identifies web technologies, plugins, JavaScript libraries, etc.

sublist3r
- Enumerator of website subdomains. Uses common web search engines

scrapy
- Application framework for web scraping, web crawling, data mining, and other web content extraction

dirbuster
- Webspider with GUI that attempts to find via brute-force all non-linked (hidden) directories and files

sf
- Spiderfoot, an OSINT website reconnaissance tool

ferret
- Cookie sniffer

hamster
- Session hijacker. It runs as a proxy, reusing cookies stolen by ferret from other clients’ sessions

fiddler
- Web debugging proxy tool, with GUI. Can be used to test the security of web applications

sslstrip
- Tool for the HTTPS Stripping attack

sqlninja
- Tool to exploit SQL injection vulnerabilities against web applications that use Microsoft SQL Server as database backend

SQLiX
- Web crawler that attempts to find SQL injection vulnerabilities on explored websites

slowloris
- Tool for the DoS Slowloris attack against web servers

slowhttptest
- Tool for testing slow HTTP DoS attacks such as Slowloris, Slow Read, R U Dead Yet, etc.

wpscan
- WordPress vulnerability scanner
snort
Network IDS/IPS with real-time traffic analysis and packet sniffing. It is configured via a customized ruleset that uses libpcap

ossec-hids
OSSEC, a HIDS with log monitoring and SIEM capabilities

tripwire
HIDS with integrity-based detection of unauthorized filesystem changes

tide
Advanced Intrusion Detection Environment, a HIDS with integrity-based detection. It makes a snapshot of the filesystem state and records it in a database, to check integrity of files at a later time

nessusd
Nessus, a well-known and comprehensive vulnerability scanner

security
Security Administrator’s Integrated Network Tool, a vulnerability scanner. Originally based on SATAN (Security Administrator Tool for Analyzing Networks)

psad
Port Scan Attack Detector. Uses iptables log messages to detect and block port scans and other malicious network traffic

john
John The Ripper, a password cracker

hashcat
Password cracker and password recovery tool

dsniff
Password sniffer, able to operate over many cleartext network protocols

hydra
Login cracker tool supporting many network protocols and password cracking techniques

medusa
Brute-force login cracker

ncrack
High-speed network authentication cracker

ophcrack-cli
Password cracker for MS Windows passwords. Uses rainbow tables

patator
Multi-purpose brute-forcer (for FTP, SSH, telnet, SMTP, DNS, MySQL, PostgreSQL, etc.)

rcrack
Hash cracker that uses rainbow tables. Part of the RainbowCrack package

rtgen
Rainbow table generator for rcrack

honeyd
Honeypot daemon. It creates virtual hosts, and simulates their networking stack and any desired network service

labrea
Honeypot for incoming IP connections. Replies to unanswered ARP requests, creating a virtual host with the related unused IP address, which then performs Layer 4 tarpitting

sshipot
SSH honeypot

artillery
Honeypot with monitoring and alerting system

honeytrap
Extensible toolkit for running and monitoring honeypots

kojoney
Low-interaction honeypot that emulates an SSH server

honeypy
Medium-interaction honeypot

cowrie
High-interaction SSH and Telnet honeypot

pentbox
Security suite including password crackers, honeypots, DoS tools, etc.

websploit
Exploit framework containing reconnaissance and attack tools for various technologies

nexphisher
Automated phishing toolkit featuring many social media websites

stegdetect
Detector of stenographic content in graphic image files

inspy
LinkedIn enumerator. Attempts to find technologies and people at a specified target company

recon-ng
Web reconnaissance framework

dog
Recon Dog, an OSINT reconnaissance tool

maltego
OSINT tool with GUI that visualizes discovered data in a graph format for link analysis
<p>| <strong>Denial of Service (DoS)</strong> | Cyberattack towards hosts or networks, aimed at preventing or reducing availability of services to legitimate users. Countermeasures: blackhole filtering aka null routes, to drop all traffic coming from the attacker. Detection via activity profiling, sequential change-point detection (Cumulative Sum algorithm), wavelet-based signal analysis of traffic's spectral components. |
| <strong>Distributed Denial of Service (DDoS)</strong> | DoS launched simultaneously from several attacking hosts (usually a group of compromised machines i.e. a botnet). |
| <strong>Distributed Reflected Denial of Service (DRDoS)</strong> | DDoS carried out by forging requests to a large number of remote hosts using the target host's spoofed source IP address. |
| <strong>Permanent Denial of Service (PDoS)</strong> | Hardware-targeted DoS which replaces the target device's firmware with a faulty one, brickling the device permanently. |
| <strong>Multi-vector attack</strong> | DoS combining volumetric, protocol, and application-layer attacks. |
| <strong>Advanced persistent threat (APT)</strong> | Stealthy attack where the attacker gains unauthorized access to a system and remains undetected for a long period. |
| <strong>Zero-day attack (0day)</strong> | Attack exploiting a software vulnerability that is still unknown or for which no fix exists yet. |
| <strong>Man-in-the-middle (MitM)</strong> | Network-based threat where the attacker inserts itself undetected in the communication channel between two legitimate parties (network-level hijacking) and then proceeds to sniff, relay, and possibly modify the traffic. Countermeasure: mutual authentication of parties. |
| <strong>Replay attack</strong> | Attack where the attacker eavesdrops on a communication, then maliciously sends again parts of a valid data transmission. Countermeasure: data tagging e.g. nonces. |
| <strong>Side channel attack</strong> | Attack based on information obtained from the implementation of a system (e.g. analysis of power consumption, timing, electromagnetic leaks, sound) and not from weaknesses in the algorithm itself (e.g. cryptanalysis, software bugs). |
| <strong>Rolling code attack</strong> | Attack against the rolling code (used itself as a defense against replay attacks) used in keyless systems. The attacker jams the signal and sniffs a first code sent by the target. As the first code did not have any effect, the target sends a second code which is sniffed too by the attacker; at the same time the attacker forwards the first code which is received by the system, but the target believes it is the effect of the second code. Later on, the attacker uses the second code to gain unauthorized access to the system. |
| <strong>Banner grabbing</strong> | Reconnaissance technique consisting in initiating a connection to the desired service and noting the software type and version mentioned in the service banner. Countermeasure: configure banners to show minimal information. |
| <strong>Username enumeration</strong> | Reconnaissance technique in which the attacker tries to determine whether a specific username exists or not in the target system, or attempts to obtain a list of valid users. Countermeasure: configure the system to show minimal information about a failed login. |
| <strong>Google hacking</strong> | Reconnaissance technique consisting in using advanced operators with specific strings (i.e. dorks) in the Google search engine to find specific versions of vulnerable web applications, misconfigurations, administration panels, sensitive files not supposed to be publicly accessible, etc. |
| <strong>Man-in-the-mobile</strong> | Infection of a mobile device with malware to bypass 2FA, as the malware relays the information to the attacker. |
| <strong>Privilege escalation</strong> | Host-based threat consisting in illegally gaining elevated access to resources that are normally protected from a program or user. |
| <strong>Confused deputy attack</strong> | Type of privilege escalation consisting in tricking a legitimate, more privileged program into misusing its authority on the system. |
| <strong>Sybil attack</strong> | Act of subverting a system by creating multiple fake identities. This may allow the attacker to e.g. acquire a disproportionate level of control over a reputation system by affecting voting results, or disrupt statistics about vehicular traffic. |</p>
<table>
<thead>
<tr>
<th>Social engineering</th>
<th>Wide range of non-technical attacks consisting in deception and psychological manipulation of the target individual into divulging confidential information or performing unwarranted actions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretexting</td>
<td>Social engineering attack where the attacker invents an elaborate scenario (i.e. a pretext) to engage the target individual under a fake identity and convince them to divulge confidential information or perform unwarranted actions.</td>
</tr>
<tr>
<td>Phishing</td>
<td>Social engineering attack aimed at obtaining sensitive information from people via a fake but legitimately-looking website controlled by the attacker. Usually perpetrated via an email message containing an obfuscated link to the malicious website.</td>
</tr>
<tr>
<td>Spear phishing</td>
<td>Personalized phishing targeted at a specific individual.</td>
</tr>
<tr>
<td>Whaling</td>
<td>Phishing targeted at a high-value individual (CEO, CISO, etc.).</td>
</tr>
<tr>
<td>Vishing</td>
<td>Phishing via VoIP.</td>
</tr>
<tr>
<td>Smishing</td>
<td>Phishing via SMS.</td>
</tr>
<tr>
<td>Tailgating Piggybacking</td>
<td>Social engineering attack in which an attacker lacking proper authorization follows an authenticated individual into the targeted restricted area.</td>
</tr>
<tr>
<td>Shoulder surfing</td>
<td>Act of getting access to sensitive information by spying an individual entering the data.</td>
</tr>
<tr>
<td>USB drop attack</td>
<td>Social engineering attack consisting in leaving a bulk of malware-infected USB flash drives in public places for people to find and use.</td>
</tr>
<tr>
<td>Baiting</td>
<td>Social engineering attack consisting in leaving a bulk of malware-infected USB flash drives in public places for people to find and use.</td>
</tr>
<tr>
<td>Dumpster diving</td>
<td>Act of searching through discarded paper documents left behind by the target organization in order to find and exploit information.</td>
</tr>
<tr>
<td>Rubber hose cryptanalysis</td>
<td>Euphemism for extracting cryptographic secrets from the target by means of coercion or violence.</td>
</tr>
<tr>
<td>Black bag cryptanalysis</td>
<td>Euphemism for obtaining cryptographic secrets from the target by breaking and entering the premises, burglary, theft, etc. This term is also used for other non-cryptanalysis methods such as keystroke logging, infection via virus or trojan horse, etc.</td>
</tr>
<tr>
<td><strong>Attack</strong></td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ping of death</td>
<td>A malformed or oversized ping packet which, when reassembled by the target host, causes a buffer overflow, crashing or infecting with malicious code the target host. Obsolete; modern TCP/IP stacks, firewalls, and IDSs easily identify and discard pings of death.</td>
</tr>
<tr>
<td>Ping flood</td>
<td>DoS in which the attacker sends a large number of ICMP Echo Request packets to the target host.</td>
</tr>
<tr>
<td>ICMP flood</td>
<td></td>
</tr>
<tr>
<td>Smurf attack</td>
<td>DRDoS in which the attacker sends a large number of ICMP Echo Request packets to a network broadcast address using the target host’s spoofed source IP address. Obsolete; by default, ICMP requests to broadcast addresses are not forwarded anymore by routers.</td>
</tr>
<tr>
<td>Fraggle attack</td>
<td>DRDoS in which the attacker sends a large number of UDP traffic to ports 7 (Echo Protocol) and 19 (CHARGEN) of multiple remote hosts, using the target host’s spoofed source IP address.</td>
</tr>
</tbody>
</table>
| SYN flood               | DoS in which the attacker sends a large number of TCP SYN packets to the target host but never responds to its SYN/ACK, hence never completing the TCP three-way handshake. This creates a large number of half-open connections which, until they time out, fill up the target host’s connection queue, preventing legitimate clients to connect to it.
Countermeasures: TCP Intercept (aka SYN Proxy or SYN cookies). |
| Teardrop                | DoS in which the attacker sends mangled IP fragments with overlapping and oversized payloads to the target host, causing it to crash. Obsolete; fixed in the Linux kernel v2.1.63.                                                                 |
| Fragmentation attack    | DoS performed by sending a large number of fragmented TCP or UDP packets to the target host, which will consume resources in reassembling and inspecting them.                                                                                                    |
| Shrew attack            | Low-rate DoS that exploits the retransmission timeout (RTO) mechanism of TCP. Performed by sending out a burst of traffic to a bottleneck router at the same time the client sends a request to the server. Hence, the router suspends the data transmission, packets are dropped during the RTO, and after the RTO the client needs to resend the lost packets, slowing down the transmission. |
| MAC spoofing            | Act of setting a spoofed MAC address in the NIC in order to divert communications to the host controlled by the attacker. Countermeasure: IP Source Guard (on switches) using the DHCP snooping binding table. |
| MAC flooding            | Dispatch of multiple Ethernet frames with different source MAC addresses to a switch. This fills up a switch’s CAM table and forces the switch to failover to hub mode (i.e. broadcasting to all switch ports), allowing the attacker to sniff all network traffic in the LAN.
Countermeasure: port security with max one MAC address per interface (on switches). |
| ARP spoofing            | Dispatch of forged ARP messages into the LAN to associate the attacker’s MAC address with another host’s (often the gateway) IP address, diverting communications to the attacker. Often the first choice of attack for the purpose of sniffing, or in preparation to MitM attacks or session hijacking attacks.
Countermeasures: Dynamic ARP Inspection (on switches) using the DHCP snooping binding table, disabling of gratuitous ARP, static IP addresses and ARP tables. On wireless networks, Client Isolation (on the wireless router) which prevents wireless clients from communicating between them. |
<p>| ARP cache poisoning     |                                                                                                           |                                                                                                           |
| ARP poisoning           |                                                                                                           |                                                                                                           |
| ARP poison routing      |                                                                                                           |                                                                                                           |
| Port stealing           | ARP spoofing aimed to associate the attacker’s MAC address with another host’s IP address on a switch’s CAM table, which will then forward packets through the wrong switch port.                                                              |
| IP spoofing             | Act of setting a spoofed IP address in the NIC so the attacker’s host can appear to be some other host. Countermeasure: direct TTL probes (however, this works only if the attacker’s host is in a different subnet). |</p>
<table>
<thead>
<tr>
<th><strong>Attacks - network</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DNS spoofing</strong></td>
</tr>
<tr>
<td><strong>DNS hijacking</strong></td>
</tr>
<tr>
<td><strong>Cybersquatting</strong></td>
</tr>
<tr>
<td><strong>Typosquatting</strong></td>
</tr>
<tr>
<td><strong>Domain sniping</strong></td>
</tr>
<tr>
<td><strong>DNS cache poisoning</strong></td>
</tr>
<tr>
<td><strong>Blind response forgery</strong></td>
</tr>
<tr>
<td><strong>DNS water torture</strong></td>
</tr>
<tr>
<td><strong>DNS amplification attack</strong></td>
</tr>
<tr>
<td><strong>DNS sinkhole attack</strong></td>
</tr>
<tr>
<td><strong>Blackhole DNS attack</strong></td>
</tr>
<tr>
<td><strong>DHCP spoofing</strong></td>
</tr>
<tr>
<td><strong>DHCP starvation</strong></td>
</tr>
<tr>
<td><strong>IRDP spoofing</strong></td>
</tr>
<tr>
<td><strong>Sinkhole attack</strong></td>
</tr>
<tr>
<td><strong>Blackhole attack</strong></td>
</tr>
<tr>
<td><strong>Packet drop attack</strong></td>
</tr>
<tr>
<td><strong>Wormhole attack</strong></td>
</tr>
<tr>
<td><strong>Man-in-the-Cloud (MitC)</strong></td>
</tr>
<tr>
<td><strong>Wardialing</strong></td>
</tr>
<tr>
<td>Attack Type</td>
</tr>
<tr>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Evil twin attack</td>
</tr>
<tr>
<td>KARMA attack</td>
</tr>
<tr>
<td>Client misassociation</td>
</tr>
<tr>
<td>Disassociation attack</td>
</tr>
<tr>
<td>Beacon flood attack</td>
</tr>
<tr>
<td>Clear channel assessment attack</td>
</tr>
<tr>
<td>Queensland attack</td>
</tr>
<tr>
<td>Chopchop attack</td>
</tr>
<tr>
<td>Key Reinstallation Attack (KRACK)</td>
</tr>
<tr>
<td>Fragmentation attack</td>
</tr>
<tr>
<td>Caffe Latte attack</td>
</tr>
<tr>
<td>Hirte attack</td>
</tr>
<tr>
<td>Wardriving</td>
</tr>
<tr>
<td>Warchalking</td>
</tr>
<tr>
<td>Attacks - Bluetooth</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td><strong>Bluejacking</strong></td>
</tr>
<tr>
<td><strong>Bluesnarfing</strong></td>
</tr>
<tr>
<td><strong>Bluebugging</strong></td>
</tr>
<tr>
<td><strong>Blueprinting</strong></td>
</tr>
<tr>
<td><strong>Bluesmacking</strong></td>
</tr>
<tr>
<td><strong>BlueBorne</strong></td>
</tr>
<tr>
<td>Attacks - hijacking</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td><strong>Network-level hijacking</strong></td>
</tr>
<tr>
<td><strong>Application-level hijacking</strong></td>
</tr>
<tr>
<td><strong>Session hijacking</strong></td>
</tr>
<tr>
<td><strong>TCP/IP hijacking</strong></td>
</tr>
<tr>
<td><strong>Blind hijacking</strong></td>
</tr>
<tr>
<td><strong>RST hijacking</strong></td>
</tr>
<tr>
<td><strong>TCP reset attack</strong></td>
</tr>
<tr>
<td><strong>UDP hijacking</strong></td>
</tr>
<tr>
<td><strong>Source routing attack</strong></td>
</tr>
<tr>
<td><strong>Session fixation</strong></td>
</tr>
<tr>
<td><strong>Session prediction</strong></td>
</tr>
<tr>
<td><strong>Session brute-forcing</strong></td>
</tr>
<tr>
<td><strong>Session riding</strong></td>
</tr>
<tr>
<td><strong>Session sidejacking</strong></td>
</tr>
<tr>
<td><strong>Session stealing</strong></td>
</tr>
<tr>
<td><strong>Session hijacking</strong></td>
</tr>
<tr>
<td><strong>万个 Lawrence</strong></td>
</tr>
<tr>
<td><strong>万千 Lawrence</strong></td>
</tr>
<tr>
<td><strong>万千 Lawrence</strong></td>
</tr>
<tr>
<td><strong>万千 Lawrence</strong></td>
</tr>
<tr>
<td><strong>万千 Lawrence</strong></td>
</tr>
<tr>
<td><strong>万千 Lawrence</strong></td>
</tr>
<tr>
<td>Attacks - web services</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td><strong>HTTP response-splitting attack</strong></td>
</tr>
<tr>
<td><strong>HTTP header injection</strong></td>
</tr>
<tr>
<td>Attack which consists in adding header response data into an input field so that the webserver splits the HTTP response into two; these can either be served to the attacker, served to a legitimate client, or discarded. Countermeasure: validation of client input. In particular, CR (%0d, \r) and LF (%0a, \n) characters should never be allowed in input.</td>
</tr>
<tr>
<td><strong>HTTP request tampering</strong></td>
</tr>
<tr>
<td>Unauthorized access to a web application obtained either by tampering with the URL query string, or by modifying the HTTP headers sent to the webserver e.g. the <strong>Referer:</strong> header which in vulnerable applications is used for access control.</td>
</tr>
<tr>
<td><strong>HTTP Parameter Pollution (HPP)</strong></td>
</tr>
<tr>
<td>Evasion technique used to bypass WAF security filters which consists in crafting an HTTP request containing multiple instances of a parameter with the same name, hence splitting the attack vector.</td>
</tr>
<tr>
<td><strong>HTTP Parameter Fragmentation (HPF)</strong></td>
</tr>
<tr>
<td>Evasion technique, often used along with HPP, which allows to reconstruct the parameter string passed in the HTTP request.</td>
</tr>
<tr>
<td><strong>Webcache poisoning</strong></td>
</tr>
<tr>
<td>Attack where the attacker uses a specially crafted request to force the webserver’s cache to flush its contents and insert a URL with infected content in the cache, which is then served to legitimate clients accessing the cache. Can be performed via an HTTP response-splitting attack.</td>
</tr>
<tr>
<td><strong>Directory traversal</strong></td>
</tr>
<tr>
<td>Unauthorized access to directories outside the webserver’s root directory, done by using repeatedly the . . . sequence in URLs.</td>
</tr>
<tr>
<td><strong>Unvalidated redirect</strong></td>
</tr>
<tr>
<td>Phishing in which the URL is that of a legitimate site but contains a redirect to the malicious site.</td>
</tr>
<tr>
<td><strong>Unvalidated forwarding</strong></td>
</tr>
<tr>
<td>Unauthorized access to a restricted webpage obtained fraudulently via an embedded forward query on the URL.</td>
</tr>
<tr>
<td><strong>CRIME</strong></td>
</tr>
<tr>
<td>Compression Ratio Info-leak Made Easy. Exploit against authentication web cookies transmitted over compressed HTTPS and SPDY connections, which results in session hijacking.</td>
</tr>
<tr>
<td><strong>BREACH</strong></td>
</tr>
<tr>
<td>Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext. Session hijacking exploit analogous to CRIME, but performed against HTTPS when using HTTP compression.</td>
</tr>
<tr>
<td><strong>HTTPS Stripping</strong></td>
</tr>
<tr>
<td><strong>SSL Stripping</strong></td>
</tr>
<tr>
<td>MitM attack which consists in hijacking the connection establishment from the target host to a remote secure webserver, then transparently downgrading all HTTPS traffic to HTTP. This attack works for web servers with redirection from HTTP to HTTPS, and only if the initial request to the webserver is sent as HTTP; then, if the client does not explicitly specify HTTPS for links, the attacker intercepts all HTTP 302 redirections and sends the client the requested content through HTTP. Countermeasure: HSTS (however, browsers not accepting HSTS cookies will still be vulnerable).</td>
</tr>
<tr>
<td><strong>Forbidden attack</strong></td>
</tr>
<tr>
<td>Attack exploiting vulnerable implementations of the TLS protocol that incorrectly reuse the same cryptographic nonce when data is encrypted. This allows the attacker to sniff the HTTPS connection and inject content.</td>
</tr>
<tr>
<td><strong>HTTP flood</strong></td>
</tr>
<tr>
<td>Layer-7 DDoS targeted at web servers. Carried out by sending a large number of HTTP GET and POST requests towards the target webserver.</td>
</tr>
<tr>
<td><strong>Slow HTTP attack</strong></td>
</tr>
<tr>
<td>Low-bandwidth DoS targeted at web servers. Carried out by keeping several connections to the target webserver open as long as possible, depleting the webserver’s connection pool. This is done by exchanging HTTP traffic at an extremely slow rate (1 byte/min or less), preventing the webserver from going into timeout for idle connection.</td>
</tr>
<tr>
<td><strong>Slow Post attack</strong></td>
</tr>
<tr>
<td>Slow HTTP attack carried out by sending correct HTTP requests and headers at an extremely slow rate.</td>
</tr>
<tr>
<td><strong>Slow Read attack</strong></td>
</tr>
<tr>
<td>Slow HTTP attack carried out by reading the webserver’s response at an extremely slow rate.</td>
</tr>
<tr>
<td><strong>Slowloris</strong></td>
</tr>
<tr>
<td>Similar to the Slow Post attack, but carried out by sending partial HTTP request and headers, while never completing the request.</td>
</tr>
<tr>
<td><strong>R U Dead Yet (RUDY)</strong></td>
</tr>
<tr>
<td>Slow HTTP attack carried out by opening concurrent POST HTTP connections and delaying sending the body of the POST request.</td>
</tr>
</tbody>
</table>
## Cross-Site Scripting (XSS)
Injection of malicious client-side scripts (e.g. via input textfields) in webpages, which are then executed when the webpage is viewed by other clients. This can lead to session hijacking, data theft, exploiting user privileges, etc. Identification of entry points for user input is the first step for exploiting a website. Countermeasures: set the HttpOnly flag in session cookies, sanitize user input.

<table>
<thead>
<tr>
<th>Non-persistent XSS</th>
<th>Reflected XSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSS attack where the malicious data provided by the attacker is used immediately by server-side scripts to display results to the targeted user only.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Persistent XSS</th>
<th>Stored XSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSS attack where the malicious data provided by the attacker is saved by the server and permanently displayed to all users visiting the website.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server-side XSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSS attack where the malicious data provided by the attacker is wholly processed server-side. Historically, this was the first kind of XSS attack.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOM-based XSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSS attack where the malicious data provided by the attacker does not affect the webserver but it is reflected fully client-side, where all the presentation logic (often JavaScript) occurs.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross-Site Flashing (XSF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSS attack where the malicious data provided by the attacker is used in some specific video playing functions and variables by Flash scripts.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross-Site Tracing (XST)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSS attack, involving the use of the HTTP TRACE method, that allows stealing cookies via Javascript.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cross-Site Request Forgery (CSRF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack which consists in having the target client unknowingly submit a maliciously crafted web request to a webpage. Can be performed via specially designed image tags, hidden forms, JavaScript functions, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server-Side Request Forgery (SSRF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack which consists in inducing the server-side application to make HTTP requests to an arbitrary domain. This results in unauthorized actions and access to data, and may lead to arbitrary command execution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URL parameter tampering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification of parameters in the URL to exploit vulnerable applications that use them, e.g. <a href="http://www.bank.com/account?id=345&amp;amount=200000">http://www.bank.com/account?id=345&amp;amount=200000</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web parameter tampering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session poisoning</td>
</tr>
<tr>
<td>Unauthorized access to a web application by exploiting (e.g. via URL parameter tampering) weak user input validation. The same term may also indicate the takeover of a session of a legitimate user e.g. by injecting malicious content, or via cookie poisoning.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cookie sniffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unauthorized access to a web application by sniffing a cookie belonging to a legitimate user and reusing it to bypass the authentication process, ending up logged in as that user.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cookie replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replay attack in which the attacker captures a cookie belonging to a legitimate user, then reuses it to perform malicious activities on behalf of that user. The attack persists until the user logs off.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cookie poisoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unauthorized access to a web application by crafting a cookie, or by sniffing and modifying a cookie belonging to a legitimate user.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cookie parameter tampering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unauthorized access to a web application by tampering with the parameters of a cookie and resubmitting it.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSI injection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code injection technique consisting in injecting scripts in webpages via SSI. SSI (Server Side Includes) are directives present on web applications which allow inserting dynamic content into an HTML page before it is loaded or visualized.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CORS attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack exploiting the Cross-Origin Resource Sharing (CORS) mechanism. CORS allows restricted resources on a webpage to be requested from another domain outside the domain from which the first resource was served; this is safer than allowing all cross-domain requests. It bypasses the Same-Origin Policy (SOP) which forbids certain cross-domain requests (e.g. Ajax).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection String Parameter Pollution (CSPP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Injection of connection string parameters into other existing parameters, often carried out by using a ; character. This can be done e.g. in connection strings for backend databases.</td>
</tr>
<tr>
<td>Attacks - generic web</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td><strong>Website defacement</strong></td>
</tr>
<tr>
<td><strong>Pharming</strong></td>
</tr>
<tr>
<td><strong>Watering hole attack</strong></td>
</tr>
<tr>
<td><strong>Framing attack</strong></td>
</tr>
<tr>
<td><strong>Clickjacking</strong></td>
</tr>
<tr>
<td><strong>UI redress attack</strong></td>
</tr>
<tr>
<td><strong>UI redressing (UIR)</strong></td>
</tr>
<tr>
<td><strong>Man-in-the-browser (MitB)</strong></td>
</tr>
</tbody>
</table>
### XML External Entity (XXE)

SSRF attack consisting in passing malicious XML input, referencing an external entity containing infected data, to an application with a weakly configured XML parser. This may cause confidential data disclosure, DoS, remote code execution, etc. XML DoS issues are a common SOA vulnerability.

### Billion laughs attack

**XML bomb**

Exponential entity expansion attack consisting in a specially crafted XML document where the root element contains a defined entity which contains e.g. 10 defined entities, each of which contains e.g. 10 defined entities, and so on. This takes up a large amount of memory and results in a DoS when the XML document is processed. Countermeasures: capping the memory allocated for an individual XML parser, treating entities symbolically and expanding them only when needed.

### XML injection

**XML poisoning**

Insertion of specially crafted XML fields in SOAP requests for web services. This may cause confidential data disclosure, DoS, remote code execution, etc.

### XPath injection

Code injection technique in which the attacker provides malicious parameters to construct the wrong XPath query and access the wrong XML node. XPath is a query language for selecting nodes from an XML document.

### Wrapping attack

**[XML]**

Attack performed during the translation of a SOAP message in the TLS layer, by intercepting the message, adding the body to the header, injecting a malicious payload in the body, and sending the message to the server as a legitimate client. The server therefore verifies the XML signature as valid. Countermeasures: XML schema validation, authenticated encryption in the XML-Enc (XML Encryption) specification.

### SQL injection (SQLI)

Code injection technique in which the attacker inserts malicious SQL statements into an input field for execution. This is done by using single quotes (') and double quotes (").

Countermeasures: user input sanitization, avoiding constructing dynamic SQL with concatenated user input values, keeping untrusted data separate from commands and queries, using least privilege account types for connection to the database.

### In-band SQL injection

SQL injection where the attacker uses the same communication channel to send the query and retrieve the result. Most common type of SQL injection. Can be:

- **Error-based**: the attacker causes the database to throw an error, e.g. by forcing a conversion type, and gains knowledge by analyzing the error message
- **Illegal query**: the attacker sends a logically incorrect query and gains knowledge by analyzing the error message
- **Union**: the attacker uses the UNION operator to get the field values from other tables; can be coupled to other operators e.g. ORDER BY to find the number of fields in a table
- **Tautology**: the attacker inserts the OR operator with a tautology so that a WHERE clause is always true, e.g. ' 1'='1
- **End-of-line comment**: the attacker uses -- to insert an end-of-line comment to make the database parser ignore the rest of the query
- **Inline comment**: the attacker uses /* */ to insert an inline comment to bypass fields

**Piggybacked query aka stacked queries**: the attacker uses a ; character to insert additional malicious queries to the original query

**Stored procedure**: the attacker input is fed to dynamic SQL statements

**Second order**: the attacker’s input is saved in a database and used later when the attacker submits a second query; requires knowledge of the application backend

### Out-of-band SQL injection

SQL injection where the attacker retrieves the result via a different channel e.g. email, HTTP, or file I/O functions. More difficult to perform.

### Blind SQL injection

**Inferential SQL injection**

SQL injection attack used when the application does not show a useful error message and hence the attacker needs to assess whether the application might be vulnerable to an SQL injection. This attack is time-intensive. Can be:

- **Time-based aka double blind**: the attacker sends a query which may be true or false embedded with sleep or benchmark functions, and gets information from the time delay that occurs in the response

**Boolean-based**: the attacker sends a query which causes the application to return a different result depending on whether the query returns true or false

**Heavy query**: the attacker sends a query that takes noticeable time to execute, e.g. a `SELECT COUNT(*)` from multiple tables

### ORM injection

SQL injection attack against a data access object model generated by ORM (Object Relational Mapping).
<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer overflow</strong></td>
<td>Attack carried out by writing data to a buffer over the buffer’s boundaries, overwriting the adjacent memory addresses. This allows the attacker to modify the target process’ address space to control process execution, crash the process, and modify variables. Countermeasure: buffer bounds checking.</td>
</tr>
<tr>
<td>Buffer overrun</td>
<td></td>
</tr>
<tr>
<td><strong>Fork bomb</strong></td>
<td>DoS attack consisting in a process that recursively replicates itself ad infinitum, depleting system resources and slowing down or crashing the system due to resource starvation.</td>
</tr>
<tr>
<td><strong>Code injection</strong></td>
<td>Attack in which the attacker inserts text in a data field that gets interpreted as code.</td>
</tr>
<tr>
<td><strong>File injection</strong></td>
<td>Code injection technique in which the attacker exploits dynamic file include mechanisms in vulnerable web applications (especially in PHP). The attacker provides a URL pointing to the malicious file, which is used by the web application instead of the intended local file.</td>
</tr>
<tr>
<td><strong>LDAP injection</strong></td>
<td>Code injection technique in which the attacker inserts malicious LDAP user parameters into an input field to get access to the LDAP database. Countermeasure: user input sanitization.</td>
</tr>
<tr>
<td><strong>Insecure deserialization attack</strong></td>
<td>Injection of malicious code into a serialized object; if the application uses a vulnerable algorithm for deserialization, the code is executed when the object is deserialized. Serialization is the act of converting an object into a format (e.g. XML, JSON, YAML) which can be written to disk, transmitted over the network, or sent to a stream (e.g. stdout).</td>
</tr>
<tr>
<td><strong>Cross-guest VM breach</strong></td>
<td>Side channel attack carried out by running a VM on the same physical host as the target VM and taking advantage of shared physical resources (e.g. CPU cache) to extract cryptographic secrets.</td>
</tr>
<tr>
<td><strong>Shrink-wrap code attack</strong></td>
<td>Attack consisting in exploiting holes in unpatched or misconfigured software (e.g. software with default insecure configuration options).</td>
</tr>
<tr>
<td><strong>Logjam</strong></td>
<td>Vulnerability affecting Diffie-Hellman key exchanges with key size from 512 to 1024 bits, allowing an attacker to downgrade a TLS connection and read and modify the exchanged data.</td>
</tr>
<tr>
<td><strong>Shellshock</strong></td>
<td>Vulnerability in the Bash shell which allows an attacker to execute arbitrary commands by exploiting the function export feature of Bash.</td>
</tr>
<tr>
<td><strong>Heartbleed</strong></td>
<td>Buffer over-read vulnerability in the OpenSSL cryptography library. The attacker client sends a heartbeat request specifying a length larger than the message payload. The server replies filling up all the allocated buffer with the contents of the active memory, which may include private data.</td>
</tr>
<tr>
<td><strong>Spectre</strong></td>
<td>Vulnerability affecting the branch prediction mechanism (done to improve performances) on microprocessors. An attacker can exploit the side effects of speculative execution, and particularly branch misprediction, to access private data.</td>
</tr>
<tr>
<td><strong>Meltdown</strong></td>
<td>Vulnerability affecting some Intel x86, IBM POWER, and ARM microprocessors. An attacker can have a process, even without proper authorization, read any memory location by exploiting a race condition between memory access and privilege check during instruction processing.</td>
</tr>
<tr>
<td>Attacks - cryptography</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Linear cryptanalysis</strong></td>
<td>Cryptanalysis based on finding affine approximations to the actions of a cipher, analyzing pairs of plaintext and the corresponding ciphertext to try to recover the encryption key.</td>
</tr>
<tr>
<td><strong>Differential cryptanalysis</strong></td>
<td>Cryptanalysis based on the analysis of how differences in the input affect the output.</td>
</tr>
<tr>
<td><strong>Integral cryptanalysis</strong></td>
<td>Cryptanalysis based on the analysis of pairs of inputs differing in only one bit.</td>
</tr>
<tr>
<td><strong>Knownplaintext attack</strong></td>
<td>Linear cryptanalysis technique where the attacker has access to some plaintext as well as the corresponding ciphertext.</td>
</tr>
<tr>
<td><strong>Chosen plaintext attack</strong></td>
<td>Cryptanalysis technique where the attacker is able to obtain the ciphertext corresponding to a plaintext of their choice.</td>
</tr>
<tr>
<td><strong>Chosen ciphertext attack</strong></td>
<td>Cryptanalysis technique where the attacker is able to obtain the plaintext corresponding to a ciphertext of their choice.</td>
</tr>
<tr>
<td><strong>Adaptive chosen plaintext attack</strong></td>
<td>Cryptanalysis technique where the attacker has access to the encryption device and is able to obtain the ciphertexts corresponding to plaintexts of their choice, making adaptive changes in the plaintext where needed.</td>
</tr>
<tr>
<td><strong>Adaptive chosen ciphertext attack</strong></td>
<td>Cryptanalysis technique where the attacker has access to the encryption device and is able to obtain the plaintexts corresponding to cipherteks of their choice, making adaptive changes in the ciphertext where needed.</td>
</tr>
<tr>
<td><strong>Non-adaptive chosen ciphertext attack</strong></td>
<td>Chosen ciphertext attack where the attacker has access to a limited set of plaintexts and ciphertexts, or has access to the device for a limited time.</td>
</tr>
<tr>
<td><strong>Related key attack</strong></td>
<td>Cryptanalysis technique where the attacker is able to obtain the ciphertext corresponding to a plaintext of their choice, encrypted with two different but similar keys.</td>
</tr>
<tr>
<td><strong>Chosen key attack</strong></td>
<td>Cryptanalysis technique where the attacker must have access to the communication channel, and obtain the plaintexts corresponding to cipherteks of their choice. Using this information, the attacker tries to recover the key by breaking an n-bit key cipher into 2^n/2 number of operations.</td>
</tr>
<tr>
<td><strong>Birthday attack</strong></td>
<td>Cryptanalysis technique that exploits the mathematics behind the birthday paradox in probability theory, i.e. the higher likelihood of hash collisions found between random attack attempts and a fixed degree of permutations.</td>
</tr>
</tbody>
</table>
| **Brute force attack** | The technique of trying every possible password or key until the correct one is found. May be very time-consuming or even infeasible.  
**Countermeasure:** long passwords and keys. |
| **Rainbow table attack** | Cracking of password hashes by using a rainbow table, i.e. a pre-computed table that caches the output of cryptographic hash functions.  
**Countermeasure:** adding a salt value to passwords before hashing. |
| **Dictionary attack** | Password cracking done by trying every entry listed in a dictionary file.  
The dictionary is a text file containing all words, names, entries with numbers and symbols added to words or replacing letters, finger rolls, etc.  
**Countermeasure:** truly random passwords. |
<p>| <strong>Hybrid attack</strong> | Password cracking combining a dictionary attack and brute force attack, done by adding numbers and symbols to the dictionary entries. |
| <strong>Meet-in-the-Middle attack</strong> | Type of known plaintext attack carried out against ciphers that use multiple keys for encryption. The attacker performs a brute force attack on one key to encrypt the plaintext and on another key to decrypt the ciphertext, trying to find an intermediate ciphertext that matches both. |
| <strong>DUHK (Don't Use Hardcoded Keys)</strong> | Cryptographic vulnerability affecting devices that use the ANSI X9.31 Random Number Generator in conjunction with a hardcoded seed key. The attacker can exploit this vulnerability to recover encryption keys from VPN connections or encrypted web sessions. |</p>
<table>
<thead>
<tr>
<th>Virus</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A <strong>virus</strong></td>
<td>is a piece of self-replicating code that attaches copies of itself to other executable programs, infecting them.</td>
</tr>
<tr>
<td>File virus</td>
<td>Infects an executable file, overwriting it.</td>
</tr>
<tr>
<td>Boot Sector virus</td>
<td>Executes its code before the target machine boots. Moves the Master Boot Record to another location on the hard disk and copies itself to the original location of the MBR.</td>
</tr>
<tr>
<td>System virus</td>
<td></td>
</tr>
<tr>
<td>Multipart virus</td>
<td>Acts both as a file virus and a Boot Sector virus.</td>
</tr>
<tr>
<td>Hybrid virus</td>
<td></td>
</tr>
<tr>
<td>FAT virus</td>
<td>Infects the File Allocation Table in FAT filesystems.</td>
</tr>
<tr>
<td>Cluster virus</td>
<td>Infects files without changing them, saving instead the virus code to the hard disk and overwriting the pointer in the directory entry; in this way, the virus code is read instead of the actual program.</td>
</tr>
<tr>
<td>Stealth virus</td>
<td></td>
</tr>
<tr>
<td>Tunneling virus</td>
<td></td>
</tr>
<tr>
<td>Sparse infector virus</td>
<td>Infects files only occasionally, upon satisfying certain conditions (time, size of the file to infect, etc.) to minimize the risk of being detected by AV software.</td>
</tr>
<tr>
<td>Encryption virus</td>
<td>Consists of a decryption module and an encrypted copy of the virus. Once the target machine is infected, the decryption module decrypts and executes the virus. The virus then infects files with a copy of the virus which is encrypted with a different key for each file. This is done to thwart signature detection methods in AV software.</td>
</tr>
<tr>
<td>Polymorphic virus</td>
<td>Reprograms itself, changing the malicious code at each infection. Consists of the encrypted virus, a decryption routine, and a mutation engine; the virus code mutates with each infection while the virus algorithm stays the same. Undetectable by AV software.</td>
</tr>
<tr>
<td>Metamorphic virus</td>
<td>Reprograms itself, rewriting the code each time it infects a new file: it inserts dead code, reorders instructions, and modifies the program control structure to avoid pattern recognition by AV software. Stealthier than a polymorphic virus.</td>
</tr>
<tr>
<td>Cavity virus</td>
<td>When infecting, overwrites empty spaces (nulls) in the original file so not to modify its size.</td>
</tr>
<tr>
<td>Camouflage virus</td>
<td>Creates a companion file with a different file extension for each file to infect; the companion file is executed first and contains the malicious code.</td>
</tr>
<tr>
<td>Companion virus</td>
<td></td>
</tr>
<tr>
<td>Shell virus</td>
<td>Wraps itself around the infected file, hosting the original program as its subroutine, so that the virus code is executed first.</td>
</tr>
<tr>
<td>TSR virus</td>
<td>Terminate and Stay Resident virus. Remains resident in memory after the infected program has terminated execution.</td>
</tr>
<tr>
<td>Macro virus</td>
<td>Written as a macro (often in VBA language), infects Microsoft Office files.</td>
</tr>
<tr>
<td>File extension virus</td>
<td>Tries to hide itself by adding a fake safe file extension (e.g. TXT) to its executable file.</td>
</tr>
<tr>
<td>Logic bomb</td>
<td>Virus that is triggered in response to a specific event.</td>
</tr>
</tbody>
</table>
### Trojan

A **Trojan** is a malicious program packed and concealed, with the help of a wrapper, inside an innocuous program.

<table>
<thead>
<tr>
<th>Trojan Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Access Trojan (RAT)</td>
<td>Provides full access to the infected host, including files, shell, screen capture, webcam, microphone, etc.</td>
</tr>
<tr>
<td>Backdoor Trojan</td>
<td>Allows bypassing the standard system authentication through IDSs and firewalls.</td>
</tr>
<tr>
<td>Botnet Trojan</td>
<td>Infects a large number of hosts (which, once compromised, become bots aka zombies) to create a botnet that can then be controlled via a Command&amp;Control center to carry out distributed attacks.</td>
</tr>
<tr>
<td>Rootkit Trojan</td>
<td>Composed of a dropper, a loader, and a rootkit; the dropper runs the loader which causes a buffer overflow, causing the rootkit to be loaded into memory.</td>
</tr>
<tr>
<td>Proxy server Trojan</td>
<td>Allows an attacker to use the infected system as a proxy to connect to the Internet.</td>
</tr>
<tr>
<td>Covert channel Trojan</td>
<td>Creates a covert channel in the data stream authorized by the network access control system, allowing the attacker to tunnel malicious traffic undetected.</td>
</tr>
</tbody>
</table>

### Rootkit

A **rootkit** is a malicious program used to gain full, administrator-level, persistent access to a system without detection.

<table>
<thead>
<tr>
<th>Rootkit Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware/firmware rootkit</td>
<td>Located in the firmware (hard disks, BIOS, etc.), creates a persistent malware image.</td>
</tr>
<tr>
<td>Bootloader-level rootkit</td>
<td>Replaces the bootloader. Can activate itself before the OS starts.</td>
</tr>
<tr>
<td>Kernel-level rootkit</td>
<td>Runs at kernel level in Ring 0 with the highest OS privileges. This is the most difficult type of rootkit to detect.</td>
</tr>
<tr>
<td>Hypervisor-level rootkit</td>
<td>Runs in Ring 1, hosting the OS of the target machine as a VM and intercepting all hardware calls made by the target OS.</td>
</tr>
<tr>
<td>Library-level rootkit</td>
<td>Patches, hooks, or replaces OS system calls with backdoored versions.</td>
</tr>
<tr>
<td>Application-level rootkit</td>
<td>Replaces application files and modifies process' behaviour by injecting malicious code.</td>
</tr>
</tbody>
</table>

### Other malware

<table>
<thead>
<tr>
<th>Malware Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worm</td>
<td>Standalone malicious program that replicates itself and executes through network connections. Usually it does not infect files, and uses just the CPU and RAM of the infected host.</td>
</tr>
<tr>
<td>Keylogger</td>
<td>Program that covertly intercepts and records all keys pressed on the keyboard. May also be hardware. The best location for it to be placed is the keyboard hardware or the OS.</td>
</tr>
<tr>
<td>Ransomware</td>
<td>Malware that encrypts files in the infected system, blocking the legitimate user from accessing them, and asks for a ransom to be paid online.</td>
</tr>
</tbody>
</table>
A firewall monitors incoming and outgoing network traffic, allowing or discarding it based on custom security rules.

Firewall architectures:

- **Bastion host**
  Defends the LAN from outside attacks. Has one public interface connected to the Internet, and one private interface connected to the LAN.

- **Screened subnet**
  Has one public interface connected to the Internet, one private interface connected to the LAN, and one interface connected to the DMZ. Designed to host servers that offer public services.

- **Multi-homed firewall**
  Composed of multiple firewalls, or one firewall with at least three NICs, which allows the protection of multiple network segments.

Firewall technologies:

- **Packet filtering**
  Filters packets according to source and destination IP address, source and destination TCP/UDP port, TCP flags, etc.

- **Circuit-level gateway**
  Forwards data between networks, making traffic appear to have originated from the gateway.

- **Application-level filtering**
  Filters traffic depending on the protocol.

- **Stateful multilayer inspection**
  Combines packet filtering, circuit-level gateway, and application-level filtering to remember the state of previous packets and determine whether session packets are legitimate.

- **Application proxy**
  Provides protection by preventing direct connections between systems on either side of the firewall, and hence by avoiding exposure of the proxied service. A client connects to the proxy firewall (stateful) which then initiates a new network connection on behalf of the request.

- **Virtual Private Network (VPN)**
  Using traffic encryption and encapsulation, provides secure access to a private network through a WAN.

- **Network Address Translation (NAT)**
  The same as in routers, remaps LAN's private IP addresses to Internet-routable public IP addresses.

Techniques for firewall reconnaissance and evasion:

- **Port scanning**
  The attacker performs a port scan to identify the firewall type and model.

- **Firewalking**
  The attacker sends a TCP/UDP packet with a TTL equal to the number of hops to the firewall plus one, to determine gateway ACL filters and perform network mapping.

- **IP address spoofing**
  The attacker spoofs the IP address of a trusted host in the network.

- **DNS poisoning**
  The attacker performs DNS poisoning, directing a legitimate user inside the firewalled perimeter to a malicious server which infects the user's host.

- **Source routing**
  The attacker sends source routed packets so that they bypass the firewall.

- **Tiny fragments**
  The attacker sends tiny fragments of outgoing packets forcing some TCP packet's header information into the next fragment, to bypass user-defined firewall rules.

- **Proxying**
  The attacker bypasses URL blacklisting by using an HTTP proxy.

- **ICMP tunneling**
  The attacker sends data in the payload portion of an ICMP Echo packet, which is usually not inspected by the firewall.

- **ACK tunneling**
  The attacker sends data in a TCP ACK packet, which is usually not inspected by the firewall as it is used as response to legitimate traffic.

- **HTTP tunneling**
  The attacker tunnels network traffic through HTTP, to bypass firewalls which only allow HTTP.

- **SSH tunneling**
  The attacker tunnels network traffic through SSH, as firewalls often allow SSH.

- **Encoded XSS**
  The attacker performs a XSS attack against a WAF, encoding the payload in ASCII or hex to avoid triggering the firewall's filters.
An IDS (Intrusion Detection System) monitors, detects, and alerts about possible intrusions (passive mode). An IPS (Intrusion Prevention System) also blocks them (active mode). An IDS can be network-based (NIDS) or host-based (HIDS).

Methods used by an IDS to detect intrusions:

**Signature recognition**
Pattern-matching of packets over signatures, generated at the network and transport layers, belonging to a known intrusion model.

**Anomaly detection**
Detection of an event outside the tolerance threshold of normal traffic and behavior.

**Protocol anomaly detection**
Detection of packets not following protocol standards.

Techniques for IDS evasion:

**Insertion attack**
The attacker obscures the exploit by inserting extra packets which will be received by the IDS but not by the target host.

**TTL attack**
Insertion attack in which the attacker sets the appropriate TTL on packets so that they will be received by the IDS but will not reach the target host.

**Invalid RST**
Insertion attack in which the attacker sends an RST packet with incorrect IP checksum, which the IDS interprets as the end of the session (and hence stops processing the traffic stream), while the target host drops the packet due to invalid checksum.

**Urgency flag**
Insertion attack in which the attacker sends a packet with the URG flag set. In such a packet the Urgent Pointer field indicates how much of the data in the segment, counting from the first byte, is urgent and should be prioritized. However, unlike the target host, some IDS ignore this field and consider the whole packet payload.

**Pre-connection SYN**
The attacker sends an initial SYN with invalid TCP checksum before the real connection is established. This desynchronizes the IDS with respect to TCP sequence numbers.

**Post-connection SYN**
The attacker sends a SYN after the connection is established. This desynchronizes the IDS with respect to TCP sequence numbers, but the target host ignores this packet as it references an already established connection. The attacker then sends an RST with the correct (for the IDS) TCP sequence number to close its connection.

**Evasion**
The attacker sends portions of the exploit in crafted packets that the IDS mistakenly does not consider. This can be done e.g. by inserting data in the payload of a TCP SYN.

**Unicode evasion**
The attacker bypasses the IDS by encoding the packets in Unicode (UTF-8 and UTF-16), which features code points for multiple representations of the same character.

**DoS**
The attacker overwhelms the IDS capacity (CPU, RAM, disk space, network bandwidth) by sending a large amount of bogus traffic.

**Obfuscation**
The attacker obscures the exploit by encoding the packet payload in a way that it will be understood by the target host but not by the IDS.

**Polymorphic shellcode**
The attacker encrypts the packet and adds the code to decrypt it inside the packet, so that the IDS cannot recognize the attack signature. Done via a buffer overflow exploit where the return memory address points to the entrance point of the decryption code. Countermeasure: checking for no-op opcodes other than 0x90.

**ASCII shellcode**
The attacker encodes the attack code in ASCII characters which translate to assembly instructions, so that the IDS cannot recognize the attack signature.

**Application-layer attack**
The attacker hides the attack code inside application-layer compressed data (audio, video, images, etc.), so that the IDS cannot recognize the attack signature.

**False positive**
The attacker sends a large amount of traffic known to trigger alerts on the IDS, hiding the real attack traffic under the noise.

**Session splicing**
The attacker splits the exploit in several small packets so that no single packet triggers the IDS in itself. The attacker can also add a delay between packets to cause the IDS to timeout before it can reassemble and check them.

**Fragmentation attack**
If the IDS has a shorter timeout than the target host for reassembling fragmented packets, the attacker sends packet fragments with a long delay; the IDS drops them while the target hosts reassembles them.

If the IDS has a longer timeout, the attacker sends mixed valid and bogus packet fragments with a short delay; the IDS gets a reassembled packet with wrong checksum and drops it, while the target host gets the reassembled exploit payload.

**Overlapping fragments**
The attacker sends the exploit fragmented in packets with overlapping TCP sequence numbers. Different OSES handle fragments differently, so the IDS may end up with a bogus packet while the target host may end up with the exploit payload.
A WIDS (Wireless Intrusion Detection System) or WIPS (Wireless Intrusion Prevention System) monitors the radio spectrum to detect, block, and alert about possible intrusions in a wireless network.

Threats detected and prevented by a WIDS or WIPS:

- Rogue Access Points
- Wireless attack tools
- DoS attacks
- MitM attacks
- Honeypots
- Evil twin attacks
- MAC spoofing
- Ad hoc networks
- Client misassociation with an AP
- Unauthorized association with an AP
- Misconfigured APs
A honeypot is a part of a computer and network system set up as a bait for attackers. It looks like a legitimate part of the site and appears to contain information or resources valuable to attackers, but in fact it is isolated and heavily monitored. It is used to identify attackers and their techniques, as well as to deflect attacks.

Types of honeypots:

**Low-interaction honeypot**
Emulates a limited number of services and applications, and provides limited interaction with the attacker. Used to collect information about attack vectors such as network probes and worm infections.

**Medium-interaction honeypot**
Emulates a real OS with services and applications, but can only respond to preconfigured commands so the attacker might notice that its functioning is limited. Allows gathering more detailed data about an attack.

**High-interaction honeypot**
Not an emulation, but a real system actually running services and applications, or a whole network architecture of systems (honeynet). As such, it may be compromised or infected by an attacker. All intrusion activities are logged, so it allows collecting a large amount of data about the attacker. A honeywall gateway allows the attacker to interact with the honeynet while limiting outbound traffic via IDS technologies, to control the attacker and preventing harm to computers outside the honeynet.

**Production honeypot**
Emulates a real production system of an organization. Used to lure attackers to trigger alerts and hence get early warnings about an attack.

**Research honeypot**
High-interaction honeypot deployed in research, military, or government organizations to get detailed information about cyberattacks in order to study exploits and vulnerabilities.

Characteristics and methods used by honeypots (which can be used to identify them):

**Layer 2 tarpit**
A tarpit is a service purposely used to delay incoming connections. A Layer 2 tarpit is used to block network penetration of an attacker, but can be detected by the use of the MAC address 00:00:0f:ff:ff:ff which acts as a blackhole for Layer 2 connections.

**Layer 4 tarpit**
In a Layer 4 tarpit, the honeypot server receives the initial SYN packet from the attacker and replies with a SYN/ACK, but then does not open a socket and forgets everything about the connection. The attacker sends an ACK, believes the TCP three-way handshake to be completed, and starts sending data which never reaches its destination. The attacker does not receive acknowledgment for the sent packets and retransmits them; the connection eventually times out. Layer 4 tarpitting can also be done by setting a TCP Window Size of 0 to block the attacker from sending further data, while keeping the connection open.

**Layer 7 tarpit**
Layer 7 tarpits are implemented by having services with a very high latency, e.g. a SMTP server which sends useless replies and waits a long time (15 secs or more) between lines.

**SYN Proxy**
As a protection against SYN flood attacks, a host applying SYN Proxy to a TCP connection responds to the initial SYN packet from the attacker with a specially crafted SYN/ACK, then waits for the ACK in response before forwarding the connection request to the server. The lack of SYN/ACK retransmissions may indicate the presence of a honeypot, especially honeyd.

**Bait and switch**
Technique consisting in redirecting all network communications to a honeypot once an intrusion attempt is detected. Attacker-side, this can be identified by the change in TCP/IP parameters (RTT, TTL, TCP timestamps, etc.)

**Fake Access Points**
Honeypots may create fake Wi-Fi APs. These APs have random SSIDs and send beacon frames but no other traffic.

**User-mode Linux (UML)**
User-mode Linux allows multiple virtual Linux kernel-based guest OSes to run as an application within a host Linux system. It is often used for honeypots, so references to UML in /proc subdirectories may indicate that the attacked system is a honeypot.