**Selinux in linux**

Introduction

QuantcastSecurity-Enhanced Linux (SELinux) is a mandatory access control (MAC) security mechanism implemented in the kernel. SELinux was first introduced in RHEL4 and significantly enhanced in RHEL5.

**The Problems**

In order to better understand why SELinux is important and what it can do for you it is easiest to look at some examples. Without SELinux enabled, discretionary access control (DAC) methods such as file permissions or access control lists (ACLs) are used to grant file access to users. Users and programs alike are allowed to grant insecure file permissions to others or gain access to parts of the system that should not otherwise be necessary for normal operation. For example:

* Administrators have no way to control users: A user could set world readable permissions on sensitive files such as ssh keys
* Processes can change security properties: A user’s mail files should be readable only by that user, but the mail client software

has the ability to change them to be world readable

* Processes inherit user’s rights: Firefox, if compromised, can read a user’s private ssh keys even though it has no reason to do so.

Essentially there are two privilege levels, root and user, and no easy way to enforce the model of least-privilege. Many processes that are launched by root later drop their rights to run as a restricted user and some processes may be run in a chroot jail, but all of these security methods are discretionary.

**The Solution**

SELinux follows the model of least-privilege. By default, everything is denied and then a policy is written that gives each element of the system (a service, program, user) only the access required to function. If a service, program or user tries to access or modify a file or resource not necessary for it to function then access is denied and the action is logged. Because SELinux is implemented within the kernel, individual applications do not need to be especially written or modified to work with SELinux. If SELinux blocks an action, this appears as just a normal “access denied” type error to the application.

**SELinux Modes**

SELinux has 3 basic modes of operation out of which **Enforcing** is set as the default mode

* **Enforcing:** The default mode which will enable and enforce the SELinux security policy on the system, denying access and logging actions
* **Permissive:** In Permissive mode, SELinux is enabled but will not enforce the security policy, only warn and log actions. Permissive mode is useful for troubleshooting SELinux issues
* **Disabled:** SELinux is turned off

The SELinux mode can be viewed and changed by using the SELinux Management GUI tool available on the Administration menu or from the command line by running ‘system-config-selinux’ (the SELinux Management GUI tool is part of the policycoreutils-gui package and is not installed by default).

Users who prefer the command line may use the ‘sestatus’ command to view the current SELinux status:

**# sestatus**

SELinux status: enabled

SELinuxfs mount: /selinux

Current mode: enforcing

Mode from config file: enforcing

Policy version: 21

Policy from config file: targeted

The ‘setenforce’ command may be used to switch between **Enforcing** and **Permissive** modes on the fly but note that these changes do not persist

through a system reboot.

To make changes persistent through a system reboot, edit the SELINUX= line in /etc/selinux/config for either ‘enforcing’, ‘permissive’, or ‘disabled’. For example, SELINUX=permissive.

[http://rhcewithvishal.files.wordpress.com/2010/06/capture.jpg?w=854&h=81](http://rhcewithvishal.files.wordpress.com/2010/06/capture.jpg)

**SELinux Policy**

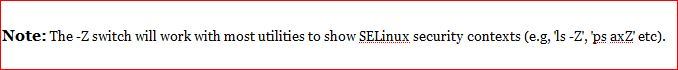
Earlier we mentioned that SELinux follows the model of least-privilege; by default everything is denied and then a policy is written that gives each element of the system only the access required to function. This description best describes the **strict** policy. However, such a policy is difficult to write that would be suitable in the wide range of circumstances that a product such as Enterprise Linux is likely to be used. The end result is that SELinux is likely to cause problems for system administrators and end users, and rather than resolve these issues system administrators are likely to just disable SELinux which defeats the purpose.

Luckily, SELinux allows different policies to be written that are interchangeable. The default policy in CentOS 4 and 5 is the **targeted** policy which “targets” and confines key system processes. In CentOS 4 only 15 defined targets existed (including httpd, named, dhcpd, mysqld) whereas in CentOS 5 this number has risen to over 200 targets. Everything else on the system runs in an unconfined domain and is unaffected by SELinux. The goal is for every process that is installed and running at boot by default to be running in a confined domain. The **targeted** policy is designed to protect as many key processes as possible without adversely affecting the end user experience and most users should be totally unaware that SELinux is even running.

**SELinux Access Control**

SELinux has 3 forms of access control:

* **Type Enforcement (TE):** Type Enforcement is the primary mechanism of access control used in the **targeted** policy
* **Role-Based Access Control (RBAC):** Based around SELinux users (not necessarily the same as the Linux user), but not used in the default **targeted** policy
* **Multi-Level Security (MLS):** Not used and often hidden in the default **targeted** policy.

All processes and files have an SELinux security context. Lets see these in action by looking at the SELinux security context of our Apache homepage, index.html: **$ ls -Z /var/www/html/index.html -rw-r–r– phil phil system\_u:object\_r:httpd\_sys\_content\_t /var/www/html/index.html** [](http://rhcewithvishal.files.wordpress.com/2010/06/capture1.jpg)

In addition to the standard file permissions and ownership, we can see the SELinux security context fields: system\_u:object\_r:httpd\_sys\_content\_t.

This is based upon user:role:type:mls. In our example above, user:role:type fields are displayed and mls is hidden. Within the default **targeted** policy, **type** is the important field used to implement Type Enforcement, in this case httpd\_sys\_content\_t.

Now lets look at the SELinux security context of the Apache web server process, httpd:

**$ ps axZ | grep httpd**

**system\_u:system\_r:httpd\_t 3234 ? Ss 0:00 /usr/sbin/httpd**

Here we see the from the type field that Apache is running under the httpd\_t type domain.

Finally, lets look at the SELinux security context of a file in our home directory:

**$ ls -Z /home/phil/myfile.txt**

**-rw-r–r– phil phil user\_u:object\_r:user\_home\_t /home/phil/myfile.txt**

where we see the type is user\_home\_t, the default type for files in a users home directory.

Access is only allowed between similar types, so Apache running as httpd\_t can read /var/www/html/index.html of type httpd\_sys\_content\_t. Because Apache runs in the httpd\_t domain, it can not access /home/phil/myfile.txt even though this file is world readable because it’s SELinux security context is not of type httpd\_t. If Apache were to be exploited, it would not be able to start any process not in the httpd\_t domain (which prevents escalation of privileges) or access any file not in an httpd\_t related domain.

**Troubleshooting SELinux**

Sooner or later you may run into situations were SELinux denies access to something and you need to troubleshoot the issue. There are a number of fundamental reasons why SELinux may deny access to a file, process or resource:

* A mislabeled file
* A process running under the wrong SELinux security context
* A bug in policy. An application requires access to a file that wasn’t anticipated when the policy was written and generates an error
* An intrusion attempt.

The first 3 we can deal with whereas the 4th case is exactly the intended behaviour.

To troubleshoot any issue, log files are key and SELinux is no different. By default SELinux log messages are written to /var/log/audit/audit.log via the Linux Auditing System (auditd) which is started by default. If auditd is not running then messages are written to /var/log/messages. SELinux log messages are labeled with the “AVC” keyword so that they can be easily filtered from other messages.

Starting with CentOS 5 the SELinux Troubleshooting tool can be used to help analyze log files converting them into a more human-readable format. The tool consists of a GUI tool for displaying messages in human-readable format and possible solutions, a desktop notification icon alerting of new issues and a daemon process (setroubleshootd) that checks for new SELinux AVC alerts and feeds the notification icon (email notifications may also be configured for those not running an X server). The SELinux Troubleshooting tool is provided by the setroubleshoot package and is installed by default. The tool may be launched from the System menu or from the command line:

**$ sealert -b**

Those not running an X server may generate human-readable reports from the command line:

**sealert -a /var/log/audit/audit.log > /path/to/mylogfile.txt**

**Relabeling Files**

The ‘chcon’ command may be used to change SELinux security context of a file or files/directories in a similar way to how chown or chmod may be used to change the ownership or standard file permissions of a file.

Lets look at some examples.

Using Apache as an example, suppose you want to change the DocumentRoot to serve web pages from a location other than the default /var/www/html directory. Suppose we create a directory (or maybe a mount point) at /html and create our index.html file there:

**# mkdir /html**

**# touch /html/index.html**

**# ls -Z /html/index.html**

**-rw-r–r– root root user\_u:object\_r:default\_t /html/index.html**

**# ls -Z | grep html**

**drwxr-xr-x root root user\_u:object\_r:default\_t html**

we see that both the directory /html and file /html/index.html have the security context type default\_t. If we start our web browser and try to view the page SELinux will deny access and log the error because the directory and file(s) have the wrong security context. We need to set the correct security context type for Apache of httpd\_sys\_content\_t:

**# chcon -v –type=httpd\_sys\_content\_t /html**

**context of /html changed to user\_u:object\_r:httpd\_sys\_content\_t**

**# chcon -v –type=httpd\_sys\_content\_t /html/index.html**

**context of /html/index.html changed to user\_u:object\_r:httpd\_sys\_content\_t**

**# ls -Z /html/index.html**

**-rw-r–r– root root user\_u:object\_r:httpd\_sys\_content\_t /html/index.html**

**# ls -Z | grep html**

**drwxr-xr-x root root user\_u:object\_r:httpd\_sys\_content\_t html**

Equally we could have set both in one go using the -R recursive switch:

**# chcon -Rv –type=httpd\_sys\_content\_t /html**

Modifying security contexts in this manner will persist between reboots unless the complete filesystem is relabeled (see later). To make the security context changes permanent, even through a complete filesystem relabel, we can use the SELinux Management Tool or the ‘semanage’ command from the command line:

**semanage fcontext -a -t httpd\_sys\_content\_t “/html(/.\*)?”**

to add a file context of type httpd\_sys\_content\_t for everything under /html.

**Restore Default Security Contexts**

The ‘restorecon’ command may be used to restore file(s) default SELinux security contexts.

Again, lets use Apache as an example. Suppose a user edits a copy of index.html in his/her home directory and moves (mv) the file to the DocumentRoot /var/www/html. Whilst the copy (cp) command will typically adopt the destination directory’s or file’s security context, move (mv) will maintain the source’s security context. We could use the ‘chcon’ command to change the security context of the file(s) in question but as the file(s) are now in the default Apache DocumentRoot (/var/www/html) we can just restore the default security contexts for that directory or file(s). To restore just the index.html file, we would use:

**# restorecon -v /var/www/html/index.html**

or to recursively restore the default security contexts for the whole directory:

**# restorecon -Rv /var/www/html**

Additionally, if we simply wanted to examine the security contexts of the /var/www/html directory to see if any files needed their security contexts restored, we can use restorecon with the -n switch to prevent any relabelling occurring:

**# restorecon -Rv -n /var/www/html**

**Relabel Complete Filesystem**

Sometimes it is necessary to relabel the complete filesystem although this should only be necessary when enabling SELinux after it has been disabled or when changing the SELinux policy from the default **targeted** policy to **strict**. To automatically relabel the complete filesystem upon reboot, do:

**# touch /.autorelabel**

**# reboot**

Sometimes a complete filesystem relabel will fail if the system has been upgraded to CentOS-5.2 with SELinux disabled, and SELinux is then enabled. If the above procedure doesn’t correctly perform a complete filesystem relabel, try issuing the ‘genhomedircon’ command first:

**# genhomedircon**

**# touch /.autorelabel**

**# reboot**

**Allowing Access to a Port**

We may want a service such as Apache to be allowed to bind and listen for incoming connections on a non-standard port. By default, the SELinux policy will only allow services access to recognized ports associated with those services. If we wanted to allow Apache to listen on tcp port 81, we can add a rule to allow that using the ‘semanage’ command:

**# semanage port -a -t http\_port\_t -p tcp 81**

A full list of ports that services are permitted access by SELinux can be obtained with:

**# semanage port -l**

**Customizing SELinux Policies**

Minor modifications to SELinux policies can be made without modifying and recompiling the policy source by setting boolean values for optional features. Such features include allowing users to share their home directories under Samba or allowing Apache to serve files from users home directories which would otherwise be denied by the SELinux policy.

There is a separate [Wiki page](http://rhcewithvishal.wordpress.com/TipsAndTricks/SelinuxBooleans) dealing with booleans.

**Which boolean do I need?**

**getsebool -a**

will show you all available booleans on your system which can be changed by you. So take a look at the list that gives you and check the booleans which might be interesting for you against the list below to see if it really does what you think it does.

**admin@forge:~$/usr/sbin/getsebool -a | grep httpd**

**allow\_httpd\_anon\_write --> off**

**allow\_httpd\_bugzilla\_script\_anon\_write --> off**

**allow\_httpd\_mod\_auth\_pam --> off**

**allow\_httpd\_nagios\_script\_anon\_write --> off**

**allow\_httpd\_squid\_script\_anon\_write --> off**

**allow\_httpd\_sys\_script\_anon\_write --> off**

**httpd\_builtin\_scripting --> on**

**httpd\_can\_network\_connect --> off**

**httpd\_can\_network\_connect\_db --> off**

**httpd\_can\_network\_relay --> off**

**httpd\_disable\_trans --> off**

**httpd\_enable\_cgi --> on**

**httpd\_enable\_ftp\_server --> off**

**httpd\_enable\_homedirs --> on**

**httpd\_rotatelogs\_disable\_trans --> off**

**httpd\_ssi\_exec --> off**

**httpd\_suexec\_disable\_trans --> off**

**httpd\_tty\_comm --> on**

**httpd\_unified --> on**

**httpd\_can\_network\_connect** looks interesting – let us check with the list below:

httpd\_can\_network\_connect (HTTPD Service):: Allow HTTPD scripts and modules to connect to the network.

Looks like it could be the one you need …

**setsebool -P httpd\_can\_network\_connect on**

will turn that on for you. Et voilà – it works.

**system-config-selinux** from the policycoreutils-gui package has the same list as the one below. So if you have a GUI available you probably are better off installing that package and making the changes there.

**For all others: Here is the :)**

**acct\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for acct daemon**

**allow\_cvs\_read\_shadow (CVS)**

**Allow cvs daemon to read shadow**

**allow\_daemons\_dump\_core (Admin)**

**Allow all daemons to write corefiles to /.**

**allow\_daemons\_use\_tty (Admin)**

**Allow all daemons the ability to use unallocated ttys.**

**allow\_execheap (Memory Protection)**

**Allow unconfined executables to make their heap memory executable. Doing this is a really bad idea. Probably indicates a badly coded executable, but could indicate an attack. This executable should be reported in bugzilla**

**allow\_execmem (Memory Protection)**

**Allow unconfined executables to map a memory region as both executable and writable, this is dangerous and the executable should be reported in bugzilla**

**allow\_execmod (Memory Protection)**

**Allow all unconfined executables to use libraries requiring text relocation that are not labeled textrel\_shlib\_t**

**allow\_execstack (Memory Protection)**

**Allow unconfined executables to make their stack executable. This should never, ever be necessary. Probably indicates a badly coded executable, but could indicate an attack. This executable should be reported in bugzilla**

**allow\_ftpd\_full\_access (FTP)**

**Allow ftpd to full access to the system**

**allow\_ftpd\_anon\_write (FTP)**

**Allow ftpd to upload files to directories labeled public\_content\_rw\_t**

**allow\_ftpd\_use\_cifs (FTP)**

**Allow ftp servers to use cifs used for public file transfer services.**

**allow\_ftpd\_use\_nfs (FTP)**

**Allow ftp servers to use nfs used for public file transfer services.**

**allow\_gpg\_execstack (Memory Protection)**

**Allow gpg executable stack**

**allow\_gssd\_read\_tmp (NFS)**

**Allow gssd to read temp directory.**

**allow\_httpd\_anon\_write (HTTPD Service)**

**Allow httpd daemon to write files in directories labeled public\_content\_rw\_t**

**allow\_httpd\_mod\_auth\_pam (HTTPD Service)**

**Allow Apache to use mod\_auth\_pam.**

**allow\_httpd\_sys\_script\_anon\_write (HTTPD Service)**

**Allow httpd scripts to write files in directories labeled public\_content\_rw\_t**

**allow\_java\_execstack (Memory Protection)**

**Allow java executable stack**

**allow\_kerberos (Kerberos)**

**Allow daemons to use kerberos files**

**allow\_mount\_anyfile (Mount)**

**Allow mount to mount any file**

**allow\_mounton\_anydir (Mount)**

**Allow mount to mount any dir**

**allow\_mplayer\_execstack (Memory Protection)**

**Allow mplayer executable stack**

**allow\_nfsd\_anon\_write (NFS)**

**Allow nfs servers to modify public files used for public file transfer services.**

**allow\_polyinstantiation (Polyinstatiation)**

**Enable polyinstantiated directory support.**

**allow\_ptrace (Compatibility)**

**Allow sysadm\_t to debug or ptrace applications**

**allow\_rsync\_anon\_write (rsync)**

**Allow rsync to write files in directories labeled public\_content\_rw\_t**

**allow\_smbd\_anon\_write (Samba)**

**Allow Samba to write files in directories labeled public\_content\_rw\_t**

**allow\_ssh\_keysign (SSH)**

**Allow ssh to run ssh-keysign**

**allow\_unconfined\_execmem\_dyntrans (Memory Protection)**

**Allow unconfined to dyntrans to unconfined\_execmem**

**allow\_user\_mysql\_connect (Databases)**

**Allow user to connect to mysql socket**

**allow\_user\_postgresql\_connect (Databases)**

**Allow user to connect to postgres socket**

**allow\_write\_xshm (XServer)**

**Allow clients to write to X shared memory**

**allow\_ypbind (NIS)**

**Allow daemons to run with NIS**

**allow\_zebra\_write\_config (Zebra)**

**Allow zebra daemon to write it configuration files**

**amanda\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for amanda**

**amavis\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for amavis**

**apmd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for apmd daemon**

**arpwatch\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for arpwatch daemon**

**auditd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for auditd daemon**

**automount\_disable\_trans (Mount)**

**Disable SELinux protection for automount daemon**

**avahi\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for avahi**

**bluetooth\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for bluetooth daemon**

**canna\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for canna daemon**

**cardmgr\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for cardmgr daemon**

**ccs\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for Cluster Server**

**cdrecord\_read\_content (User Privs)**

**Allow cdrecord to read various content. nfs, samba, removable devices, user temp and untrusted content files**

**ciped\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ciped daemon**

**clamd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for clamd daemon**

**clamscan\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for clamscan**

**clvmd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for clvmd**

**comsat\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for comsat daemon**

**courier\_authdaemon\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for courier daemon**

**courier\_pcp\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for courier daemon**

**courier\_pop\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for courier daemon**

**courier\_sqwebmail\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for courier daemon**

**courier\_tcpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for courier daemon**

**cpucontrol\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for cpucontrol daemon**

**cpuspeed\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for cpuspeed daemon**

**cron\_can\_relabel (Cron)**

**Allow system cron jobs to relabel filesystem for restoring file contexts.**

**crond\_disable\_trans (Cron)**

**Disable SELinux protection for crond daemon**

**cupsd\_config\_disable\_trans (Printing)**

**Disable SELinux protection for cupsd backend server**

**cupsd\_disable\_trans (Printing)**

**Disable SELinux protection for cupsd daemon**

**cupsd\_lpd\_disable\_trans (Printing)**

**Disable SELinux protection for cupsd\_lpd**

**cvs\_disable\_trans (CVS)**

**Disable SELinux protection for cvs daemon**

**cyrus\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for cyrus daemon**

**dbskkd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dbskkd daemon**

**dbusd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dbusd daemon**

**dccd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dccd**

**dccifd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dccifd**

**dccm\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dccm**

**ddt\_client\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ddt daemon**

**devfsd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for devfsd daemon**

**dhcpc\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dhcpc daemon**

**dhcpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dhcpd daemon**

**dictd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dictd daemon**

**direct\_sysadm\_daemon (Admin)**

**Allow sysadm\_t to directly start daemons**

**disable\_evolution\_trans (Web Applications)**

**Disable SELinux protection for Evolution**

**disable\_games\_trans (Games)**

**Disable SELinux protection for games**

**disable\_mozilla\_trans (Web Applications)**

**Disable SELinux protection for the web browsers**

**disable\_thunderbird\_trans (Web Applications)**

**Disable SELinux protection for Thunderbird**

**distccd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for distccd daemon**

**dmesg\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dmesg daemon**

**dnsmasq\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dnsmasq daemon**

**dovecot\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for dovecot daemon**

**entropyd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for entropyd daemon**

**fcron\_crond (Cron)**

**Enable extra rules in the cron domain to support fcron.**

**fetchmail\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for fetchmail**

**fingerd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for fingerd daemon**

**freshclam\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for freshclam daemon**

**fsdaemon\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for fsdaemon daemon**

**ftpd\_disable\_trans (FTP)**

**Disable SELinux protection for ftpd daemon**

**ftpd\_is\_daemon (FTP)**

**Allow ftpd to run directly without inetd**

**ftp\_home\_dir (FTP)**

**Allow ftp to read/write files in the user home directories**

**global\_ssp (Admin)**

**This should be enabled when all programs are compiled with ProPolice/SSP stack smashing protection. All domains will be allowed to read from /dev/urandom.**

**gpm\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for gpm daemon**

**gssd\_disable\_trans (NFS)**

**Disable SELinux protection for gss daemon**

**hald\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for hal daemon**

**hide\_broken\_symptoms (Compatibility)**

**Do not audit things that we know to be broken but which are not security risks**

**hostname\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for hostname daemon**

**hotplug\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for hotplug daemon**

**howl\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for howl daemon**

**hplip\_disable\_trans (Printing)**

**Disable SELinux protection for cups hplip daemon**

**httpd\_builtin\_scripting (HTTPD Service)**

**Allow HTTPD to support built-in scripting**

**httpd\_can\_network\_connect\_db (HTTPD Service)**

**Allow HTTPD scripts and modules to network connect to databases.**

**httpd\_can\_network\_connect (HTTPD Service)**

**Allow HTTPD scripts and modules to connect to the network.**

**httpd\_can\_network\_relay (HTTPD Service)**

**Allow httpd to act as a relay.**

**httpd\_disable\_trans (HTTPD Service)**

**Disable SELinux protection for httpd daemon**

**httpd\_enable\_cgi (HTTPD Service)**

**Allow HTTPD cgi support**

**httpd\_enable\_ftp\_server (HTTPD Service)**

**Allow HTTPD to run as a ftp server**

**httpd\_enable\_homedirs (HTTPD Service)**

**Allow HTTPD to read home directories**

**httpd\_rotatelogs\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for httpd rotatelogs**

**httpd\_ssi\_exec (HTTPD Service)**

**Allow HTTPD to run SSI executables in the same domain as system CGI scripts.**

**httpd\_suexec\_disable\_trans (HTTPD Service)**

**Disable SELinux protection for http suexec**

**httpd\_tty\_comm (HTTPD Service)**

**Unify HTTPD to communicate with the terminal. Needed for handling certificates.**

**httpd\_unified (HTTPD Service)**

**Unify HTTPD handling of all content files.**

**hwclock\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for hwclock daemon**

**i18n\_input\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for i18n daemon**

**imazesrv\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for imazesrv daemon**

**inetd\_child\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for inetd child daemons**

**inetd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for inetd daemon**

**innd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for innd daemon**

**iptables\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for iptables daemon**

**ircd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ircd daemon**

**irqbalance\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for irqbalance daemon**

**iscsid\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for iscsi daemon**

**jabberd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for jabberd daemon**

**kadmind\_disable\_trans (Kerberos)**

**Disable SELinux protection for kadmind daemon**

**klogd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for klogd daemon**

**krb5kdc\_disable\_trans (Kerberos)**

**Disable SELinux protection for krb5kdc daemon**

**ktalkd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ktalk daemons**

**kudzu\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for kudzu daemon**

**locate\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for locate daemon**

**lpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for lpd daemon**

**lrrd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for lrrd daemon**

**lvm\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for lvm daemon**

**mailman\_mail\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for mailman**

**mail\_read\_content (Web Applications)**

**Allow evolution and thunderbird to read user files**

**mdadm\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for mdadm daemon**

**monopd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for monopd daemon**

**mozilla\_read\_content (Web Applications)**

**Allow the mozilla browser to read user files**

**mrtg\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for mrtg daemon**

**mysqld\_disable\_trans (Databases)**

**Disable SELinux protection for mysqld daemon**

**nagios\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for nagios daemon**

**named\_disable\_trans (Name Service)**

**Disable SELinux protection for named daemon**

**named\_write\_master\_zones (Name Service)**

**Allow named to overwrite master zone files**

**nessusd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for nessusd daemon**

**NetworkManager\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for NetworkManager**

**nfsd\_disable\_trans (NFS)**

**Disable SELinux protection for nfsd daemon**

**nfs\_export\_all\_ro (NFS)**

**Allow NFS to share any file/directory read only**

**nfs\_export\_all\_rw (NFS)**

**Allow NFS to share any file/directory read/write**

**nmbd\_disable\_trans (Samba)**

**Disable SELinux protection for nmbd daemon**

**nrpe\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for nrpe daemon**

**nscd\_disable\_trans (Name Service)**

**Disable SELinux protection for nscd daemon**

**nsd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for nsd daemon**

**ntpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ntpd daemon**

**oddjob\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for oddjob**

**oddjob\_mkhomedir\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for oddjob\_mkhomedir**

**openvpn\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for openvpn daemon**

**pam\_console\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for pam daemon**

**pegasus\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for pegasus**

**perdition\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for perdition daemon**

**portmap\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for portmap daemon**

**portslave\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for portslave daemon**

**postfix\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for postfix**

**postgresql\_disable\_trans (Databases)**

**Disable SELinux protection for postgresql daemon**

**pppd\_can\_insmod (pppd)**

**Allow pppd daemon to insert modules into the kernel**

**pppd\_disable\_trans (pppd)**

**Disable SELinux protection for pppd daemon**

**pppd\_disable\_trans (pppd)**

**Disable SELinux protection for the mozilla ppp daemon**

**pppd\_for\_user (pppd)**

**Allow pppd to be run for a regular user.**

**pptp\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for pptp**

**prelink\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for prelink daemon**

**privoxy\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for privoxy daemon**

**ptal\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ptal daemon**

**pxe\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for pxe daemon**

**pyzord\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for pyzord**

**quota\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for quota daemon**

**radiusd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for radiusd daemon**

**radvd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for radvd daemon**

**rdisc\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for rdisc**

**readahead\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for readahead**

**read\_default\_t (Admin)**

**Allow programs to read files in non-standard locations default\_t**

**read\_untrusted\_content (Web Applications)**

**Allow programs to read untrusted content without relabel**

**restorecond\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for restorecond**

**rhgb\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for rhgb daemon**

**ricci\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ricci**

**ricci\_modclusterd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ricci\_modclusterd**

**rlogind\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for rlogind daemon**

**rpcd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for rpcd daemon**

**rshd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for rshd**

**rsync\_disable\_trans (rsync)**

**Disable SELinux protection for rsync daemon**

**run\_ssh\_inetd (SSH)**

**Allow ssh to run from inetd instead of as a daemon**

**samba\_enable\_home\_dirs (Samba)**

**Allow Samba to share users home directories**

**samba\_share\_nfs (Samba)**

**Allow Samba to share nfs directories**

**allow\_saslauthd\_read\_shadow (SASL authentication server)**

**Allow sasl authentication server to read /etc/shadow**

**saslauthd\_disable\_trans (SASL authentication server)**

**Disable SELinux protection for saslauthd daemon**

**scannerdaemon\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for scannerdaemon daemon**

**secure\_mode (Admin)**

**Do not allow transition to sysadm\_t, sudo and su effected**

**secure\_mode\_insmod (Admin)**

**Do not allow any processes to load kernel modules**

**secure\_mode\_policyload (Admin)**

**Do not allow any processes to modify kernel SELinux policy**

**sendmail\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for sendmail daemon**

**setrans\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for setrans**

**setroubleshootd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for setroublesoot daemon**

**slapd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for slapd daemon**

**slrnpull\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for slrnpull daemon**

**smbd\_disable\_trans (Samba)**

**Disable SELinux protection for smbd daemon**

**snmpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for snmpd daemon**

**snort\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for snort daemon**

**soundd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for soundd daemon**

**sound\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for sound daemon**

**spamassassin\_can\_network (Spam Assassin)**

**Allow Spam Assasin daemon network access**

**spamd\_disable\_trans (spam Protection)**

**Disable SELinux protection for spamd daemon**

**spamd\_enable\_home\_dirs (spam Protection)**

**Allow spamd to access home directories**

**spammassasin\_can\_network (spam Protection)**

**Allow spammassasin to access the network**

**speedmgmt\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for speedmgmt daemon**

**squid\_connect\_any (Squid)**

**Allow squid daemon to connect to the network**

**squid\_disable\_trans (Squid)**

**Disable SELinux protection for squid daemon**

**ssh\_keygen\_disable\_trans (SSH)**

**Disable SELinux protection for ssh daemon**

**ssh\_sysadm\_login (SSH)**

**Allow ssh logins as sysadm\_r:sysadm\_t**

**staff\_read\_sysadm\_file (Admin)**

**Allow staff\_r users to search the sysadm home dir and read files such as ~/.bashrc**

**stunnel\_disable\_trans (Universal SSL tunnel)**

**Disable SELinux protection for stunnel daemon**

**stunnel\_is\_daemon (Universal SSL tunnel)**

**Allow stunnel daemon to run as standalone, outside of xinetd**

**swat\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for swat daemon**

**sxid\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for sxid daemon**

**syslogd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for syslogd daemon**

**system\_crond\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for system cron jobs**

**tcpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for tcp daemon**

**telnetd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for telnet daemon**

**tftpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for tftpd daemon**

**transproxy\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for transproxy daemon**

**udev\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for udev daemon**

**uml\_switch\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for uml daemon**

**unlimitedInetd (Admin)**

**Allow xinetd to run unconfined, including any services it starts that do not have a domain transition explicitly defined.**

**unlimitedRC (Admin)**

**Allow rc scripts to run unconfined, including any daemon started by an rc script that does not have a domain transition explicitly defined.**

**unlimitedRPM (Admin)**

**Allow rpm to run unconfined.**

**unlimitedUtils (Admin)**

**Allow privileged utilities like hotplug and insmod to run unconfined.**

**updfstab\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for updfstab daemon**

**uptimed\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for uptimed daemon**

**use\_lpd\_server (Printing)**

**Use lpd server instead of cups**

**use\_nfs\_home\_dirs (NFS)**

**Support NFS home directories**

**user\_canbe\_sysadm (User Privs)**

**Allow user\_r to reach sysadm\_r via su, sudo, or userhelper. Otherwise, only staff\_r can do so.**

**user\_can\_mount (Mount)**

**Allow users to execute the mount command**

**user\_direct\_mouse (User Privs)**

**Allow regular users direct mouse access only allow the X server**

**user\_dmesg (User Privs)**

**Allow users to run the dmesg command**

**user\_net\_control (User Privs)**

**Allow users to control network interfaces also needs USERCTL=true**

**user\_ping (User Privs)**

**Allow normal user to execute ping**

**user\_rw\_noexattrfile (User Privs)**

**Allow user to r/w noextattrfile FAT, CDROM, FLOPPY**

**user\_rw\_usb (User Privs)**

**Allow users to rw usb devices**

**user\_tcp\_server (User Privs)**

**Allow users to run TCP servers bind to ports and accept connection from the same domain and outside users disabling this forces FTP passive mode and may change other protocols**

**user\_ttyfile\_stat (User Privs)**

**Allow user to stat ttyfiles**

**use\_samba\_home\_dirs (Samba)**

**Allow users to login with CIFS home directories**

**uucpd\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for uucpd daemon**

**vmware\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for vmware daemon**

**watchdog\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for watchdog daemon**

**winbind\_disable\_trans (Samba)**

**Disable SELinux protection for winbind daemon**

**write\_untrusted\_content (Web Applications)**

**Allow web applications to write untrusted content to disk implies read**

**xdm\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for xdm daemon**

**xdm\_sysadm\_login (XServer)**

**Allow xdm logins as sysadm\_r:sysadm\_t**

**xend\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for xen daemon**

**xen\_use\_raw\_disk (XEN)**

**Allow xen to read/write physical disk devices**

**xfs\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for xfs daemon**

**xm\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for xen constrol**

**ypbind\_disable\_trans (NIS)**

**Disable SELinux protection for ypbind daemon**

**yppasswdd\_disable\_trans (NIS)**

**Disable SELinux protection for NIS Password Daemon**

**ypserv\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for ypserv daemon**

**ypxfr\_disable\_trans (NIS)**

**Disable SELinux protection for NIS Transfer Daemon**

**zebra\_disable\_trans (SELinux Service Protection)**

**Disable SELinux protection for zebra daemon**

**httpd\_use\_cifs (HTTPD Service)**

**Allow httpd to access samba/cifs file systems.**

**httpd\_use\_nfs (HTTPD Service)**

**Allow httpd to access nfs file systems.**

**samba\_domain\_controller (Samba)**

**Allow samba to act as the domain controller, add users, groups and change passwords**

**samba\_export\_all\_ro (Samba)**

**Allow Samba to share any file/directory read only**

**samba\_export\_all\_rw (Samba)**

**Allow Samba to share any file/directory read/write**

**webadm\_manage\_users\_files (HTTPD Service)**

**Allow httpd to access nfs file systems.**

**webadm\_read\_users\_files (HTTPD Service)**

**Allow httpd to access nfs file systems.**

[**Security Enhanced Linux SELinux**](http://computernetworkingnotes.com/manage-system-security/security-enhanced-linux-selinux.html)

SELinux was Developed by the National Security Agency (NSA), it adds protection for different files, applications, processes, and so on. On the Red Hat exams, you are expected to work with SELinux.

The first objective is fundamental to SELinux

**Set enforcing/permissive modes for SELinux**

The next objective requires that you understand the SELinux contexts defined for different files and processes.

**List and identify SELinux file and process contexts**

The next objective require that you are able to restore the default file contexts

**Restore default file contexts**

The last objective require that you configure boolean setting.

**Use boolean settings to modify system SELinux settings**

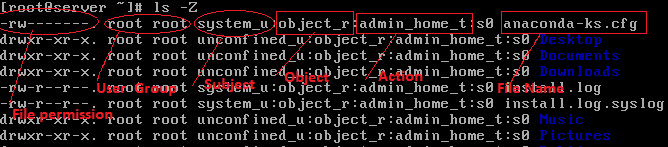
In this article we would start from the fundamental of SELinux.

**Understanding SELinux**

SELinux can be quite complex. So we would start from basic. Before you start working with SELinux you should understated the terminology used in SELinux. Let's start with some of the basics concept:

* **subject :-** subject is a command, process or application witch want to access any linux file.
* **object :-** object is a linux file or services.
* **action :-** an action is what may be done by the subject to the object.

Each file, folder, and service has an associated label that contains all three contexts.



**File Contexts :-** SELinux uses four different contexts to enforce security:

1. user[take it as subject]
2. role[understand it as object]
3. domain (also known type, this is action)
4. level (new from RHEL6 this level represents the sensitivity level of a file or directory).

for contexts you could use more restrictive values but for RHCE exam you should only focus on following contexts values.

**Important context values for RHCE Exam**

|  |  |  |
| --- | --- | --- |
| **Contexts** | **Values** | **Description** |
| User: | unconfined\_u | Unprotected user |
| system\_u | System user |
| user\_u | Normal user |
| Role: | object\_r | File |
| system\_r | Users and processes |
| Domain: | unconfined\_r | Unprotected file or process |

Take a example of sshd service check the SELinux labels



* The first field you see here is system\_u, which, you can tell from the table , is a system user.
* The second field contains system\_r, which again you can reference to see that it is a user or, in this case, a process.
* The third field shows sshd\_t as the domain.

The domain is simply a way of categorizing which contexts can do to one another. Let's take a another example of domain context

selinux domian

From output you could see

user[subject] system\_u (a system user)

role[object] object\_r(a file)

domain[action] etc\_t

Any service that has access to the etc\_t domain is able to access this file. Beside root only system services have access to the /etc directory, so a domain of etc\_t makes sense.

Now you have basic understanding of SELinux context.

**What does Security-enhanced Linux give me that standard linux can't?**  
The Security-enhanced Linux kernel enforces mandatory access control policies that confine user programs and system [**servers**](http://kyapoocha.com/linux-interview-questions/what-does-security-enhanced-linux-give-me-that-standard-linux-cant/) to the minimum amount of privilege they require to do their jobs. When confined in this way, the ability of these user programs and system daemons to cause harm when compromised (via buffer overflows or misconfigurations, for example) is reduced or eliminated. This confinement mechanism operates independently of the traditional Linux access control mechanisms. It has no concept of a “root” super-user, and does not share the well-known shortcomings of the traditional [**Linux**](http://kyapoocha.com/linux-interview-questions/what-does-security-enhanced-linux-give-me-that-standard-linux-cant/) security mechanisms (such as a dependence on setuid/setgid binaries).The [**security**](http://kyapoocha.com/linux-interview-questions/what-does-security-enhanced-linux-give-me-that-standard-linux-cant/) of an unmodified Linux system depends on the correctness of the kernel, all the privileged applications, and each of their configurations. A problem in any one of these areas may allow the compromise of the entire system. In contrast, the security of a modified system based on the Security-enhanced Linux kernel depends primarily on the correctness of the kernel and its security policy configuration. While problems with the correctness or configuration of [**applications**](http://kyapoocha.com/linux-interview-questions/what-does-security-enhanced-linux-give-me-that-standard-linux-cant/) may allow the limited compromise of individual user programs and system daemons, they do not pose a threat to the security of other user programs and system daemons or to the security of the system as a whole.

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text1&action=editWidget)

**What is SELINUX ?**

**SELinux  
Security-enhanced** [**Linux**](http://kyapoocha.com/linux-interview-questions/what-is-selinux/) **is a research prototype of the Linux® kernel and a number of utilities with enhanced security functionality designed simply to demonstrate the value of mandatory access controls to the Linux community and how such controls could be added to Linux. The Security-enhanced Linux kernel contains new architectural components originally developed to improve the** [**security**](http://kyapoocha.com/linux-interview-questions/what-is-selinux/) **of the Flask operating system. These architectural components provide general support for the enforcement of many kinds of mandatory access control policies, including those based on the concepts of Type Enforcement®, Role-based Access Control, and Multi-level Security.**

<!--[if !supportEmptyParas]--> <!--[endif]-->

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text2&action=editWidget)

**What is the most graceful way to get to run level single user mode?**

**The most graceful way is to use the command init s.  
If you want to shut everything down before going to single user mode then do init 0 first and from the ok prompt do a boot -s.**

<!--[if !supportEmptyParas]--> <!--[endif]-->

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text3&action=editWidget)

**Write a command to find all of the files which have been accessed with in the last 10 days ?**  
<!--[if !supportLineBreakNewLine]-->  
<!--[endif]-->

**The following command will find all of the files which have been accessed within the last 10 days  
find / -type f -atime -10 > December.files**

**This command will find all the files under root, which is ‘/’, with file type is file. ‘-atime -30′ will give all the files accessed less than 10 days ago. And the output will put into a file call Monthname.files.**

<!--[if !supportEmptyParas]--> <!--[endif]-->

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text4&action=editWidget)

**What is the main advantage of creating links to a file instead of copies of the file?**

**The main advantage is not really that it saves disk space (though it does that too) but, rather, that a change of permissions on the file is applied to all the link access points. The link will show permissions of lrwxrwxrwx but that is for the link itself and not the access to the file to which the link points. Thus if you want to change the permissions for a command, such as su, you only have to do it on the original. With copies you have to find all of the copies and change permission on each of the copies.**

<!--[if !supportEmptyParas]--> <!--[endif]-->

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text5&action=editWidget)

**What is LILO ?**  
  
**LILO stands for** [Linux](http://kyapoocha.com/linux-interview-questions/what-is-lilo/) **boot loader. It will load the MBR, master boot record, into the memory, and tell the system which partition and hard drive to boot from.**

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text6&action=editWidget)

**What is CVS ?**

**CVS is Concurrent Version** [System](http://kyapoocha.com/linux-interview-questions/what-is-cvs/)**. It is the front end to the RCS revision control system which extends the notion of revision control from a collection of files in a single directory to a hierarchical collection of directories consisting of revision controlled files. These directories and files can be combined together to form a software release.  
There are some useful commands that are being used very often. They are**

**cvs checkout  
cvs update  
cvs add  
cvs remove  
cvs commit**

<!--[if !supportEmptyParas]--> <!--[endif]-->

[<!--[if !vml]--><!--[endif]-->](http://www.blogger.com/rearrange?blogID=753244944686289476&widgetType=Text&widgetId=Text7&action=editWidget)

1. **How do you list the files in an UNIX directory while also showing hidden files?** ls -ltra
2. **How do you execute a UNIX command in the background?** Use the “&”.
3. **What UNIX command will control the default file permissions when files are created?** umask
4. **Explain the read, write, and execute permissions on a UNIX directory.** Read allows you to see and list the directory contents. Write allows you to create, edit and delete files and subdirectories in the directory. Execute gives you the permissions to run programs or shells from the directory.
5. **What is the difference between a soft link and a hard link?** A symbolic (soft) linked file and the targeted file can be located on the same or different file system while for a hard link they must be located on the same file system.
6. **Give the command to display space usage on the UNIX file system.** df -lk
7. **Explain iostat, vmstat and netstat.** iostat reports on terminal, disk and tape I/O activity. vmstat reports on virtual memory statistics for processes, disk, tape and CPU activity. netstat reports on the contents of network data structures.
8. **How would you change all occurrences of a value using VI?** %s/(old value)/(new value)/g
9. **Give two UNIX kernel parameters that effect an Oracle install.** SHMMAX & SHMMNI
10. **Briefly, how do you install Oracle software on UNIX?** Basically, set up disks, kernel parameters, and run orainst.

<!--[if !supportEmptyParas]--> <!--[endif]-->