

Learn more in our sysadmin's guide to SELinux, by Alex Callejas: <https://red.ht/2zpWppY>

CONCEPTS

<p>SELinux = LABELING system</p> <p>Every process, file, directory, system object has a LABEL.</p> <p>Policy rules control access between labeled processes and labeled objects.</p> <p>The kernel enforces these rules.</p>	<p>Labeling → files, process, ports, etc. (system objects)</p> <p>Type enforcement → Isolates processes from each other based on types</p>
---	--

LABELING

<p>Label format:</p> <p>user:role:type:level (optional)</p>	<p>user → identity known to the policy authorized for a specific set of roles and a specific MLS/MCS range</p> <p>role → attribute of RBAC, serves as an intermediary between domains and SELinux users</p> <p>type → attribute of type enforcement, defines a domain for processes and a type for files</p> <p>level → attribute of MLS/MCS, pair of levels, written as lowlevel-highlevel if the levels differ, or lowlevel if the levels are identical</p>
---	---

TYPE ENFORCEMENT

Targeted:	Processes that are targeted run in a confined domain, and processes that are not targeted run in an unconfined domain
Multi-level security (mls):	Control processes (domains) based on the level of the data they will be using
Multi-category security (mcs):	Protects like processes from each other (like VMs, OpenShift Gears, SELinux sandboxes, containers, etc.)

SELINUX MODES @ BOOT

<p>Kernel parameters:</p> <p>enforcing=0 → boot in permissive mode</p> <p>selinux=0 → kernel to not load any part of the SELinux infrastructure</p> <p>autorelabel=1 → forces the system to relabel</p>	<p>If you need to relabel the entire system:</p> <p># touch /.autorelabel</p> <p># reboot</p> <p>If the system labeling contains a large amount of errors, you might need to boot in permissive mode for the autorelabel to succeed.</p>
--	--

SELINUX STATES

		CHECK STATUS:		
enforcing	SELinux security policy is enforced	Configuration file:	Check if SELinux is enabled:	# getenforce
permissive	SELinux prints warnings instead of enforcing	/etc/selinux/config	SELinux status tool:	# sestatus
disabled	No SELinux policy is loaded		Enable/disable SELinux (temporarily):	# setenforce [110]

EXAMPLE OF LABELING: APACHE WEB SERVER			CHECK/CREATE/MODIFY SELINUX CONTEXTS/LABELS:
Binary	<code>/usr/sbin/httpd</code>	<code>httpd_exec_t</code>	Many commands accept the argument <code>-Z</code> to view, create, and modify context: - ls -Z - id -Z - ps -Z - netstat -Z - cp -Z - mkdir -Z Contexts are set when files are created based on their parent directory's context (with a few exceptions). RPMs can set contexts as part of installation.
Configuration directory	<code>/etc/httpd</code>	<code>httpd_config_t</code>	
Logfile directory	<code>/var/log/httpd</code>	<code>httpd_log_t</code>	
Content directory	<code>/var/www/html</code>	<code>httpd_sys_content_t</code>	
Startup script	<code>/usr/lib/systemd/system/httpd.service</code>	<code>httpd_unit_file_d</code>	
Process running	<code>/usr/sbin/httpd -DFOREGROUND</code>	<code>httpd_t</code>	
Ports (netstat -tulpnZ)	80/tcp, 443/tcp	<code>httpd_t</code>	
Port type (semanage port -l)	80, 81, 443, 488, 8008, 8009, 8443, 9000	<code>http_port_t</code>	

TROUBLESHOOTING

SELinux tools:	<code># yum -y install setroubleshoot setroubleshoot-server</code>		← Reboot or restart auditd after you install	
Logging:	<code>/var/log/messages</code>	<code>/var/log/audit/audit.log</code>	<code>/var/lib/setroubleshoot/setroubleshoot_database.xml</code>	
journalctl	List all logs related to setroubleshoot:	<code># journalctl -t setroubleshoot --since=14:20</code>		
	List all logs related to a particular SELinux label:	<code># journalctl _SELINUX_CONTEXT=system_u:system_r:policykit_t:s0</code>		
ausearch	Look for SELinux errors in the audit log:	<code># ausearch -m AVC,USER_AVC,SELINUX_ERR -ts today -i</code>		
	Search for SELinux AVC messages for a particular service:	<code># ausearch -m avc -c httpd -i</code>		
Edit/modify labels (semanage)	know the label:	<code># semanage fcontext -a -t httpd_sys_content_t 'srv/myweb(/.*)?'</code>		
	know the file with the equivalent labeling:	<code># semanage fcontext -a -e /srv/myweb /var/www</code>		
	Restore the context (for both cases):	<code># restorecon -vR /srv/myweb</code>		
Edit/modify labels (chcon)	know the label:	<code># chcon -t httpd_system_content_t /var/www/html/index.html</code>	Note: If you move instead of copy a file, the file keeps its original context.	
	know the file with the equivalent labeling:	<code># chcon --reference /var/www/html/ /var/www/html/index.html</code>		
	Restore the context (for both cases):	<code># restorecon -vR /var/www/html/index.html</code>		
Add new port to service:	<code># semanage port -a -t http_port_t -p tcp 8585</code>		← SELinux needs to know	
Booleans	Booleans allow parts of SELinux policy to be changed at runtime without any knowledge of SELinux policy writing.			
To see all booleans:	<code># getsebool -a</code>	To see the description of each one:	<code># semanage boolean -l</code>	
To set a boolean execute:	<code># setsebool [boolean] [1 0]</code>	To configure it permanently, add -P:	Example : <code># setsebool httpd_enable_ftp_server 1 -P</code>	