SELinux使用说明

# 系统文件权限的管理DAC(传统) 和 MAC(SELinux)

总结：DAC是以用户为出发点来管理权限的  
      MAC是以程序为出发点来管理权限的

## 自主式存取控制, DAC(Discretionary Access Control, DAC)

传统的文件权限与帐号关系，简单理解DAC就是rwx！因此，当某个程序想要对文件进行存取时， 系统就会根据该程序的拥有者/群组，并比对文件的权限，若通过权限检查，就可以存取该文件了。

【注意】：各种权限设定对 root 是无效的。

DAC的缺点：

* root 具有最高的权限：如果不小心某个程序被它人取得， 且该程序属于 root 的权限，那么这支程序就可以在系统上进行任何资源的存取！真是要命！
* 使用者可以取得程序来变更文件资源的存取权限：如果你不小心将某个目录的权限设定为 777 ，由于对任何人的权限会变成 rwx ，因此该目录就会被任何人所任意存取！

## 委任式存取控制, MAC

以政策规则订定特定程序读取特定文件。他可以针对特定的程序与特定的文件资源来进行权限的控管！ 也就是说，即使你是 root ，那么在使用不同的程序时，你所能取得的权限并不一定是 root ， 而得要看当时该程序的设定而定。如此一来，我们针对控制的『主体』变成了『程序』而不是使用者。 此外，这个主体程序也不能任意使用系统文件资源，因为每个文件资源也有针对该主体程序设定可取用的权限！ 如此一来，控制项目就细的多了！但整个系统程序那么多、文件那么多，一项一项控制可就没完没了！ 所以 SELinux 也提供一些预设的政策 (Policy) ，并在该政策内提供多个规则 (rule) ，让你可以选择是否启用该控制规则！

在委任式存取控制的设定下，我们的程序能够活动的空间就变小了！举例来说， WWW 伺服器软体的达成程序为 httpd 这支程式， 而预设情况下， httpd 仅能在 /var/www/ 这个目录底下存取文件，如果 httpd 这个程序想要到其他目录去存取资料时， 除了规则设定要开放外，目标目录也得要设定成 httpd 可读取的模式 (type) 才行喔！限制非常多！ 所以，即使不小心 httpd 被 cracker 取得了控制权，他也无权浏览 /etc/shadow 等重要的设定档喔！

【理解】  
以前：root--->启动httpd---->httpd可以访问系统任何文件  
现在：root--->启动httpd---->httpd只能访问/var/www/目录(这是MAC)规则的约束：如果httpd想要访问其他目录，那么必须满足两个条件：DAC的rwx + MAC的规则。

## SELinux 是透过 MAC 的方式来控管程序

在selinux中，系统的文件，目录，设备甚至端口都作为对象；用于运行的程序当做主体。主体访问对象的检测过程：DAC+MAC权限。使用selinux后，所有的进程和文件都被标记为一种类型，类型定义了进程的操作域，同时定义了文件的类型。每个进程被限制在独立的域中，由selinux策略来定义进程与文件资源以及进程与进程之间的访问权限。传统的DAC访问控制使用的是用户id和用户组id来进行权限判断。Selinux则是居于多个有效的安全标记判断权限，例如：selinux用户，角色，类型，级别。

* 主体 (Subject)：SELinux 主要想要管理的就是程序，主体= process；
* 对象(Object)：主体程序将访问的文件，目标=文件；
* 政策 (Policy)：由于程序与文件数量庞大，因此 SELinux 依据服务来制订基本的存取安全性政策。这些政策内包含有详细的规则 (rule) 来指定不同的服务开放某些资源的存取与否。主要包含两种：targeted：针对网路服务限制较多，针对本机限制较少，是预设的政策；strict：完整的 SELinux 限制，限制方面较为严格。建议使用预设 targeted 政策即可。

# SElinux配置文件

vi /etc/selinux/config

|  |
| --- |
| # This file controls the state of SELinux on the system.  # SELINUX= can take one of these three values:  # enforcing - SELinux security policy is enforced.  # permissive - SELinux prints warnings instead of enforcing.  # disabled - No SELinux policy is loaded.  SELINUX=disabled  # SELINUXTYPE= can take one of these two values:  # targeted - Targeted processes are protected,  # mls - Multi Level Security protection.  SELINUXTYPE=targeted |

## 配置详解

###配置文件有效行数为以下两行

#enforcing:表示强制开启，selinux会拦截非法资源的访问，并记录日志

#permissive:表示警告模式，访问无权限的文件时候，会记录日志但不会拦截

# disabled:表示禁用SElinux，一般从禁用变为开启的修改要重启服务器生效

SELINUX=enforcing

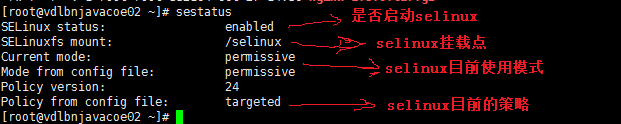
#目前仅有 targeted 与 mls（mls对系统中所有进程进行控制，会导致基本命#令都会失败）

SELINUXTYPE= targeted

## 使用命令查看和修改selinux参数

### 查看selinux详细信息

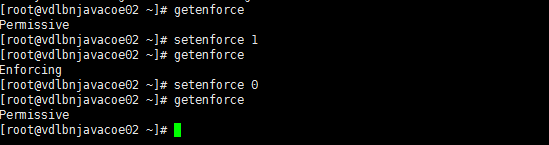
sestatus



### 查看和修改selinux模式

setenforce/getenforce

setenforce 0/1 #0代表permissive，1代表enforceing



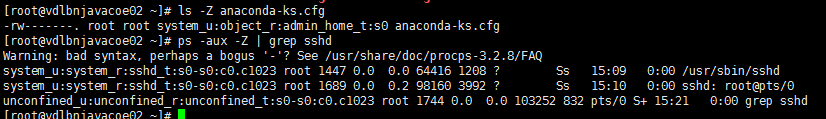
注意：要想让selinux修改永久生效必须修改配置文件/etc/selinux/config

# SElinux安全上下文

SElinux会为每一个文件添加安全信息标签：selinux用户，角色，类型，级别。用这些信息作为selinux访问控制的依据。SELinux安全上下文包括：

用户：角色：类型：级别

## 查看文件或者程序的安全上下文

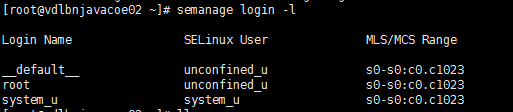
ls -Z 查看文件安全上下文/ps –aux –Z 查看进程安全上下文  


## 安全上下文详解

### SELinux用户

yum -y install policycoreutils-python

用户身份是通过SELinux策略授权特定角色的集合的账户身份，每个系统账户都通过策略映射到一个SELinux用户。通过root身份运行命令semanage login –l 命令查看系统账户和SElinux账户之间的映射关系：

* user\_u-: 普通用户登录系统后预设；
* system\_u-： 开机过程中系统进程的预设；
* root-： root登录后预设；

### SElinux角色

SELinux部分采用居于角色的访问控制RBAC模型，而角色是RBAC的重要属性。SELinux账户被授予特定的角色，而角色被授予操控特定的域。

* 文件与目录的role：通常是object\_r；
* 程序的role：通常是system\_r；
* 用户的role：targeted policy为system\_r；
* strict policy为sysadm\_r，staff\_r，user\_r

### SELinux类型

类型定义进程的域和文件的类型。而SELinux策略定义了何种类型主体访问何种类型的对象的资源。只有SELinux策略明确允许，访问才能被接受。

[root@master oracle]# ll -Zd /usr/sbin/httpd  
-rwxr-xr-x  root root system\_u:object\_r:httpd\_exec\_t   /usr/sbin/httpd  
【说明】：既然说type很重要，/usr/sbin/httpd的type是httpd\_exec\_t。 这说明httpd程序对该文件有执行权限！相当于文件上的x权限！！

### SElinux级别

级别是MLS和MCS的属性，一个MLS范围是一对级别，书写格式：低级别-高级别，如果两个级别是一致的也可以显示一个级别。s0-s0与s0是一样的。

## Selinux日志排错

grep setroublesheet /var/log/messages

sealert -l \*\*\*-\*\*\*

# 修改安全上下文

## chcon命令

描述：修改文件SELinux安全上下文

用法：chcon [选项] [-u selinux用户] [-r 角色] [-l 范围] [-t 类型] 文件

或者chcon [选项] --reference =参考文件 文件

选项：

* -u：修改用户属性
* -r：修改用户角色
* -l：修改范围
* -t：修改类型属性
* -R：递归所有子文件和子文件夹

chcon -R -t admin\_home\_t /root

#上面解释为使用smb进程能够访问/tmp/abc目录而设定的安全上下文。

chcon -R -t samba\_share\_t /tmp/abc

## semanage命令(策略管理工具)

描述：SELinux策略管理工具

用法：semanage fcontext [-S sotre] –{a|d|m|l|n|D|} [-frst] file\_spec

选项：

* -a,--add：添加预设安全上下文
* -d,--delete：删除指定的预设安全上下文
* -D,--deleteall：删除所有的语塞自定义上下文
* -m,--modify：修改指定预设安全上下文
* -l,--list：显示预设安全上下文
* -n,--noheading：不显示头部信息

semanage fcontext –l #显示所有目录和其安全上下文信息

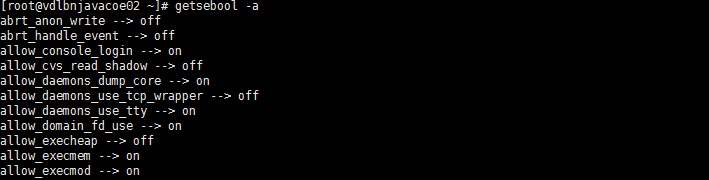
restorecon 文件 #还原文件的预设安全上下文

# SELinux 的常用使用

## 查看和修改布尔值(获取本机selinux策略值)

### 查看布尔值

getsebool –a | grep xxx或者使用semanage boolean -l



说明：selinux的设置一般通过两个部分完成的，一个是安全上下文，另一个是策略，策略值是对安全上下文的补充。

### 修改布尔值

setsebool -P name on/off

-P 是永久性设置，否则重启之后又恢复预设值。

setsebool -P allow\_ftpd\_anon\_write on

## SElinux与公共目录共享

如果ftp,samba,web都访问共享目录的话，该文件的安全上下文应为：

public\_content\_t

public\_content\_rw\_t

semanage fcontext –R –a –t public\_content\_t "目录"

semanage fcontext –R –a –t public\_content\_rw\_t "目录"

restorecon –F –R –v 目录

#服务的策略的bool值，应根据具体情况做相应的修改。

## httpd相关的selinux

### 布尔值

|  |
| --- |
| httpd\_can\_network\_connect\_db --> off  httpd\_enable\_cgi --> on  httpd\_use\_cifs --> off  allow\_httpd\_mod\_auth\_pam --> off  httpd\_use\_nfs --> off  httpd\_can\_sendmail --> off  httpd\_can\_network\_memcache --> off |

### 安全上下文

|  |
| --- |
| /var/cache缓存目录资源：httpd\_cache\_t  Apache配置文件：httpd\_config\_t  Apache日志文件资源：httpd\_log\_t  Httpd代理内容资源：httpd\_squid\_content\_t  http系统资源：httpd\_sys\_content\_t  可读写httpd系统资源：httpd\_sys\_rw\_content\_t |

chcon -R -t httpd\_sys\_content\_t /home/html

setsebool -P allow\_ftpd\_anon\_write = 1

setsebool -P allow\_httpd\_anon\_write = 1

setsebool -P allow\_<协议名>\_anon\_write = 1

#关闭selinux对httpd的保护

httpd\_disable\_trans=0

## ftp相关的selinux

### 布尔值

|  |
| --- |
| allow\_ftpd\_anon\_write --> off  allow\_ftpd\_full\_access --> off  allow\_ftpd\_use\_cifs --> off  allow\_ftpd\_use\_nfs --> off  ftp\_home\_dir --> off  ftpd\_connect\_db --> off  ftpd\_use\_passive\_mode --> off  httpd\_enable\_ftp\_server --> off  tftp\_anon\_write --> off |

### 安全上下文

|  |
| --- |
| /etc/目录下的ftp文档：ftpd\_etc\_t  控制ftp程序仅在ftpd\_t域下运行：ftpd\_exec\_t  控制ftp程序仅在ftpd\_initrc\_t域中运行：ftpd\_initrc\_exec\_t  ftp锁数据文件：ftpd\_lock\_t  ftp在/tmp目录下生成的临时文件：ftpd\_tmp\_t |

chcon -R -t public\_content\_t /var/ftp

chcon -R -t public\_content\_rw\_t /var/ftp/incoming

setsebool -P allow\_ftpd\_anon\_write =1

## mysql相关的selinux

### 布尔值

allow\_user\_mysql\_connect --> off

mysql\_connect\_any --> off

### 安全上下文

|  |
| --- |
| mysqld数据库文件：mysqld\_db\_t  /etc/目录下的mysql文件：mysqld\_etc\_t  控制mysql程序仅在mysqld\_t域中运行:mysql\_exec\_t  控制mysql程序仅在mysqld\_initrc\_t域中运行:mysqld\_initrc\_t  ftp在/tmp目录中生成临时文件:ftpd\_tmp\_t  控制mysql程序仅在mysqld\_safe\_t域中运行:mysqld\_safe\_exec\_t  mysql存储在/tmp目录下的临时文件:mysqld\_tmp\_t  mysql存储在/var/run下的文件:mysql\_var\_run\_t |

## nfs相关的selinux

### 布尔值

|  |
| --- |
| allow\_ftpd\_use\_nfs --> off  cobbler\_use\_nfs --> off  git\_system\_use\_nfs --> off  httpd\_use\_nfs --> off  qemu\_use\_nfs --> on  rsync\_use\_nfs --> off  samba\_share\_nfs --> off  sanlock\_use\_nfs --> off  sge\_use\_nfs --> off  use\_nfs\_home\_dirs --> on  virt\_use\_nfs --> off  xen\_use\_nfs --> off |

### 安全上下文

|  |
| --- |
| nfs数据文件:nfs\_t  控制nfs程序仅在ftsd\_t域中运行:nfsd\_exec\_t  控制nfs程序仅在nfsd\_initrc\_exec\_t域中执行:nfsd\_initrc\_exec\_t  设置文件为只读NFS文件:nfsd\_ro\_t  设置文件为可读写NFS文件:nfsd\_rw\_t |

selinux对nfs的限制好像不是很严格，默认状态下，不对 nfs的安全上下文进行标记，而且在默认状态的策略下，nfs的目标策略允许nfs\_export\_all\_ro  
nfs\_export\_all\_ro  
nfs\_export\_all\_rw 值为0  
所以说默认是允许访问的。但是如果共享的是/home/abc的话，需要打开相关策略对home的访问。  
setsebool -P use\_nfs\_home\_dirs boolean 1  
getsebool use\_nfs\_home\_dirs

## samba相关的selinux

### 布尔值

|  |
| --- |
| samba\_create\_home\_dirs --> off  samba\_domain\_controller --> off  samba\_enable\_home\_dirs --> off  samba\_export\_all\_ro --> off  samba\_export\_all\_rw --> off  samba\_portmapper --> off  samba\_run\_unconfined --> off  samba\_share\_fusefs --> off  samba\_share\_nfs --> off  sanlock\_use\_samba --> off  use\_samba\_home\_dirs --> off  virt\_use\_samba --> off |

### 安全上下文

|  |
| --- |
| Samba存储在/etc/目录下的文件：samba\_etc\_t  Samba存储在/tmp/目录下的文件:samba\_net\_tmp\_t  Samba存储在/var/目录下的文件:samba\_var\_t  Samba密码文件:samba\_secrets\_t  Samba共享文件: samba\_share\_t  Samba日志文件: samba\_log\_t  控制samba程序仅在samba\_initrc\_t域运行: samba\_initrc\_exec\_t  控制samba程序仅在samba\_net-t域中运行：samba\_net\_exec\_t |

1．samba共享的文件必须用正确的selinux安全上下文标记。

chcon -R -t samba\_share\_t /home/smb

如果共享/home/abc，需要设置整个主目录的安全上下文。

2．修改策略(只对主目录的策略的修改)

# setsebool -P samba\_export\_all\_ro on

setsebool -P samba\_enable\_home\_dirs=1

setsebool -P allow\_smbd\_anon\_write=1

getsebool -a | grep samba

samba\_enable\_home\_dirs -->on

allow\_smbd\_anon\_write --> on /\*允许匿名访问并且可写\*/