比如，内核打印这个提示

type=1400 audit(32.939:25): avc: denied { open } for pid=2592 comm="chmod" path="/dev/block/mmcblk0p25" dev="tmpfs" ino=6494 scontext=u:r:init\_shell:s0 tcontext=u:object\_r:block\_device:s0 tclass=blk\_file permissive=1

当然,这种信息有时太长了，以致于在minicom中出现了信息丢失的情况，可按参考http://blog.csdn[**.NET**](http://lib.csdn.net/base/dotnet)/mike8825/article/details/49422727里的方法解决。

我们可以遵循这个方法，从头开始寻找关键对象，然后调整一下顺序，生成一条语句,最后将该语句填写到.te文件(external/sepolicy/\*.te)中即可。

denied { open }             u:r:init\_shell:s0            u:object\_r:block\_device:s0       tclass=blk\_file

             A                               B                                              C                                       D  
    
             B                               C                                              D                                       A

**allow init\_shell  block\_device:blk\_file open;**

这条语句表示允许init\_shell域中的block\_device进程打开block\_device类型的块设备文件。

有时候会遇到编译该规则失败,这也许就是neverallow语句做怪了。

neverallow用来检查安全策略文件中是否有违反该项规则的allow语句

如external/sepolicy/netd.te文件中，语句

**neverallow netd dev\_type:blk\_file { read write };**

表示永远不允许netd域中的进程读写dev\_type类型的块设备文件，这时只需屏蔽该语句即可。

当然,在调试阶段，可在终端上运行如下命令获取SELinux的状态和临时关闭SELinux

setenforce 0                  ##设置SELinux 成为permissive模式（SELinux开启，但对违反selinux规则的行为只记录，不会阻止）

setenforce 1                  ##设置SELinux 成为enforcing模式 (SELinux开启)

getenforce                     ##获取SELinux状态(permissive,enforcing,disabled)