[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ENGINEER/DAY02/CASE/01/index.html" \l "page_top_case)

# NSD ENGINEER DAY02

1. [案例1：启用SELinux保护](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ENGINEER/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：自定义用户环境](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ENGINEER/DAY02/CASE/01/index.html" \l "case2)
3. [案例3：配置firewalld防火墙](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/ENGINEER/DAY02/CASE/01/index.html" \l "case3)

## 1 案例1：启用SELinux保护

### 1.1 问题

本例要求为虚拟机 server0、desktop0 配置SELinux：

1. 确保 SELinux 处于强制启用模式
2. 在每次重新开机后，此设置必须仍然有效

### 1.2 方案

SELinux，Security-Enhanced Linux：是由美国NSA国家安全局提供的一套基于内核的增强的强制安全保护机制，针对用户、进程、文档标记安全属性并实现保护性限制。

SELinux安全体系直接集成在Linux内核中，包括三种运行模式：

* disabled：彻底禁用，内核在启动时不加载SELinux安全体系
* enforcing：强制启用，内核加载SELinux安全体系，并强制执行保护策略
* permissive：宽松模式，内核加载SELinux安全体系，只记录不执行

执行getenforce可以查看当前所处的模式。

在disabled模式与enforcing、permissive模式之间切换时，需要重新启动Linux系统；而在enforcing模式与permissive模式之间切换时，并不需要重启，可以直接执行setenforce 1|0操作。

### 1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：调整当前的SELinux运行模式

1）查看当前模式

1. [root@server0 ~]# getenforce
2. Permissive                                     //表示当前为宽松模式

若上述操作显示的结果为Disabled，表示SELinux机制已被禁用，只能通过步骤修改固定配置后再重启；若显示的结果为Enforcing，表示已经处于强制启用模式。

2）切换为enforcing强制启用模式

如果在操作1）中显示的结果为Permissive，则执行以下操作切换为强制启用：

1. [root@server0 ~]# setenforce 1                 //强制启用
2. [root@server0 ~]# getenforce                     //确认切换结果
3. Enforcing

如果在操作1）中显示的结果为Disabled，则无法使用setenforcing命令：

1. [root@desktop0 ~]# getenforce
2. Disabled
3. [root@desktop0 ~]# setenforce 1
4. setenforce: SELinux is disabled

步骤二：为SELinux运行模式建立固定配置

1）修改配置文件/etc/selinux/config

1. [root@server0 ~]# vim /etc/selinux/config
2. SELINUX=enforcing
3. .. ..

2）重启验证结果

1. [root@server0 ~]# reboot
2. .. ..
3. [root@server0 ~]# getenforce
4. Enforcing

## 2 案例2：自定义用户环境

### 2.1 问题

本例要求为系统 server0 和 desktop0 创建自定义命令，相关说明如下：

1. 自定义命令的名称为 qstat
2. 此自定义命令将执行以下操作：/bin/ps -Ao pid,tt,user,fname,rsz
3. 此自定义命令对系统中的所有用户都有效

### 2.2 方案

命令别名：为一个复杂的命令行建立一个更加简短的命令字，方便重复使用。

基本管理操作：

* 定义别名：alias 别名='复杂的命令行'
* 查看别名：alias、alias 别名
* 取消别名：unalias 别名、unalias -a

用户登录初始化文件：

* 全局配置：/etc/bashrc、
* 用户自定义配置：~/.bashrc

### 2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：为主机server0添加别名qstat

1）为所有用户添加初始化命令

1. [root@server0 ~]# vim /etc/bashrc
2. .. ..
3. alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'

2）验证别名qstat是否生效

1. [root@server0 ~]# exit                                     //退出
2. logout
3. Connection to server0 closed.
4. [kiosk@foundation0 ~]$ ssh -X root@server0                 //重登录
5. Last login: Sat Nov 26 15:30:15 2016 from 172.25.0.250
6. [root@server0 ~]# alias qstat                             //可查到别名
7. alias qstat='/bin/ps -Ao pid,tt,user,fname,rsz'
8. [root@server0 ~]# qstat                                     //且此别名正常可用
9. PID TT USER COMMAND RSZ
10. 1 ? root systemd 6548
11. 2 ? root kthreadd 0
12. 3 ? root ksoftirq 0

步骤二：为主机desktop0添加别名qstat

操作与步骤一相同。

## 3 案例3：配置firewalld防火墙

### 3.1 问题

本例要求为两个虚拟机 server0、desktop0配置防火墙策略：

1. 允许从172.25.0.0/24网段的客户机访问 server0、desktop0 的任何服务
2. 在172.25.0.0/24网络中的系统，访问 server0 的本地端口5423将被转发到80
3. 上述设置必须永久有效

### 3.2 方案

Linux的防火墙体系根据所在的网络场所区分，提供了预设的安全区域：

* public：仅允许访问本机的sshd等少数几个服务
* trusted：允许任何访问
* block：阻塞任何来访请求
* drop：丢弃任何来访的数据包
* ……

新增防火墙规则的位置包括：

* 运行时（runtime）：仅当前有效，重载防火墙后失效
* 永久（permanent）：静态配置，需要重载防火墙才能生效

本地端口转发（端口1 --> 端口2）：

* 从客户机访问防火墙主机的 端口1 时，与访问防火墙的 端口 2 时等效
* 真正的网络应用服务其实在 端口2 提供监听

### 3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：采取“默认全允许，仅拒绝个别”的防护策略

1）启用防火墙服务

1. [root@server0 ~]# systemctl restart firewalld
2. [root@server0 ~]# systemctl enable firewalld

2）将默认区域设置为trusted

1. [root@server0 ~]# firewall-cmd --get-default-zone                 //修改前
2. public
3. [root@server0 ~]# firewall-cmd --set-default-zone=trusted         //修改操作
4. success
5. [root@server0 ~]# firewall-cmd --get-default-zone             //修改后
6. trusted

步骤二：封锁指定的IP网段

1）添加永久配置“阻塞来自网段172.34.0.0/24的任何访问”

1. [root@server0 ~]# firewall-cmd --permanent --zone=block --add-source=172.34.0.0/24
2. success

2）重载防火墙

1. [root@server0 ~]# firewall-cmd --reload
2. success

3）检查运行时规则

1. [root@server0 ~]# firewall-cmd --list-all --zone=block
2. block
3. interfaces:
4. sources: 172.34.0.0/24
5. services:
6. ports:
7. masquerade: no
8. forward-ports:
9. icmp-blocks:
10. rich rules:

步骤三：实现5423-->80端口转发

1）针对80端口部署测试应用

快速搭建一个测试网站：

1. [root@server0 ~]# yum -y install httpd                    //装包
2. .. ..
3. [root@server0 ~]# vim /var/www/html/index.html             //部署测试网页
4. test site.
5. [root@server0 ~]# systemctl restart httpd                 //起服务

从客户端访问，确认测试网页：

1. [root@desktop0 ~]# yum -y install elinks
2. .. ..
3. [root@desktop0 ~]# elinks -dump http://server0.example.com/
4. test site.

2）配置5423-->80端口转发策略

1. [root@server0 ~]# firewall-cmd --permanent --zone=trusted --add-forward-port=port=5423:proto=tcp:toport=80                             //添加永久配置
2. success
3. [root@server0 ~]# firewall-cmd --reload                     //重载服务
4. Success
5. [root@server0 ~]# firewall-cmd --list-all                 //确认运行时规则
6. trusted (default, active)
7. interfaces: eth1 eth2 eth0 team0
8. sources:
9. services:
10. ports:
11. masquerade: no
12. forward-ports: port=5423:proto=tcp:toport=80:toaddr=
13. icmp-blocks:
14. rich rules:

3）验证端口转发策略

从desktop0上访问server0的5423端口，与访问server0的80端口效果一样：

1. [root@desktop0 ~]# elinks -dump http://server0.example.com:5423/
2. test site.
3. [root@desktop0 ~]# elinks -dump http://server0.example.com/
4. test site.