Security-01

服务安全与监控

## 如何保护1台服务的安全

技术手段：系统安全，服务安全，数据安全，网络安全

非技术手段：管理制度实现 社会工程学

# Linux基本防护

## 用户账户安全

### /etc/skel #用户的初始环境文件

### /etc/login.defs #添加系统帐号时的默认配置文件

### 设置帐号有效期

### 使用chage工具

— d 0 ，强制修改密码,首次登录修改密码

]# useradd admin2

]# echo 123456 | passwd --stdin admin2

]# chage -d 0 admin2

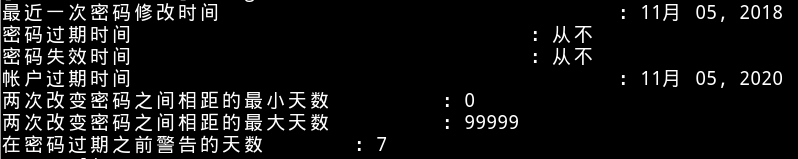
— -E yyyy-mm-dd ，指定失效日期(-1 取消)

]# chage -l admin2



]# chage -E 2020-11-05 admin2

]# chage -l admin2



]# ls /home/admin2 -a

. .. .bash\_logout .bash\_profile .bashrc .mozilla

]# cp cluster.tar.gz /etc/skel/

]# useradd admin3

]# ls /home/admin3 -a

. .. .bash\_logout .bash\_profile .bashrc cluster.tar.gz .mozilla

### 帐号的锁定和解锁

使用passwd命令

-l 锁定 、 -u 解锁 、 -s 看状态

]# useradd admin4

]# passwd -S admin4 //默认是锁定的,空密码无法登录

admin4 LK 2018-11-05 0 99999 7 -1 (密码已被锁定。)

]# echo 123456 | passwd --stdin admin4

]# passwd -S admin4 //设置密码之后，此时查看窗台是解锁状态

admin4 PS 2018-11-05 0 99999 7 -1 (密码已设置，使用 SHA512 算法。)

]# passwd -l admin4 //锁定账户，无法登录

锁定用户 admin4 的密码 。

passwd: 操作成功

]# passwd -S admin4 //查看状态

admin4 LK 2018-11-05 0 99999 7 -1 (密码已被锁定。)

]# passwd -u admin4 //解锁账户

解锁用户 admin4 的密码。

passwd: 操作成功

强制定期修改密码

配置文件/etc/login.defs //对新建用户有效

主要控制属性

— pass\_max\_days

— pass\_min\_days

— pass\_warn\_age

伪装登录提示

配置文件/etc/issue 、 /etc/issue.net

- 分别使用于本地、远程登录

- 默认会提示内核、系统等版本信息

]# cat /etc/issue

\S

Kernel \r on an \m

修改/etc/issue的文件内容即可

### 对于文件系统来说，文件系统也可以通过添加额外属性来提高性能与安全性

]# cat /etc/fstab

/dev/vda1 /boot xfs defaults,noexec 0 0

/dev/vda3 /home xfs defaults,noatime 0 0

注释：noexec属性可以让分区下的所有程序都不可以执行，包括病毒和木马

noatime让分区下的所有文件都不再更新atime时间，atime时间为文件的访问时间

## 文件系统安全

### 程序和服务控制

禁用非必要的系统服务

- 使用systemctl(REDHAT7) 、chkconfig(Redhat6)工具

]# chkconfig httpd on //设置开机自启动

注意：正在将请求转发到“systemctl enable httpd.service”。

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

]# chkconfig httpd off //设置开机不启动

]# chconfig httpd list //查看装态

### 锁定/解锁保护文件

EXT3/EXT4的文件属性控制

- chattr(修改)、lsattr(查看)

+ 、- 、 =控制方式

- 属性i：不可变(immutable)

- 属性a：仅可追加(append only)

]# lsattr /etc/passwd

---------------- /etc/passwd

]# chattr +a /etc/passwd

]# useradd admin55 //因为加了a属性，所以无法用户信息的配置文件

useradd：无法打开 /etc/passwd

]# lsattr /etc/passwd

-----a---------- /etc/passwd

]# chattr -a /etc/passwd

加a属性之后，只有用追加的方式才可写入数据

]# echo "admin5:x:1004:1004::/home/admin5:/bin/bash" >> /etc/passwd

]# chattr +i /etc/passwd

]# mv /etc/passwd /root/

mv: 无法将"/etc/passwd" 移动至"/root/passwd": 不允许的操作

]# lsattr /etc/passwd

----i----------- /etc/passwd

加i选项之后不能对文档中的数据进行修改，只能拷贝

# 用户切换与提权

## su切换用户身份

切换用户身份：ssh远程管理、运维测试

提升执行权限：管理权限细分

## su切换的基本用法

### Substitute User,换人

- 快速切换为指定的其他用户

- 普通用户执行时，须验证目标用户的口令

- root执行时，无须验证口令

### 命令格式

- 用法1：su [-] [目标用户]

- 用法2：su [-] -c “命令” [目标用户]

### 从普通用户切换为root,并登录新shell环境

- 执行su - ，或者 su - root

- 不指定目标用户时，默认视为root

[bob@client50 ~]$ whoami

bob

[bob@client50 ~]$ su -

密码： //验证root用户的口令

[root@client50 ~]# whoami

root

### root以指定的普通用户身份执行任务

- 以用户tom的身份创建目录

- 以用户tom的身份执行管理员操作会出错

[root@client50 ~]# su - tom -c 'mkdir /home/tom/test'

[root@client50 ~]# ll -d /home/tom/test/

drwxrwxr-x. 2 tom tom 6 11月 6 09:19 /home/tom/test/

[root@client50 ~]# su - tom -c 'systemctl restart sshd'

Error creating textual authentication agent:

### 分析su切换的使用情况

安全日志/var/log/secure

- 记录su验证、Shell开启与关闭

]# tail -f /var/log/secure

Nov 5 11:27:49 client50 su: pam\_unix(su-l:session): session opened for user admin2 by root(uid=0)

Nov 5 11:30:22 client50 su: pam\_unix(su-l:session): session closed for user admin2

oppend：su切换登入成功

closed：su会话断开成功

## sudo提权的基本用法

### Super or another Do,超级执行

- 管理员预先为用户设置执行许可

- 被授权用户有权执行授权的命令，验证自己的口令

### 命令格式

- 用法1：sudo 特权命令

- 用法2：sudo [-u 目标用户] 特权命令

### 查看自己的sudo授权

- sudo -l

### 以用户tom的权限新建一个文件夹

- sudo -u tom mkdir /tmp/mydir

## 配置sudo授权

修改方法：推荐visudo,也可以使用vim /etc/sudoers

授权记录格式

- 用户 主机列表=命令列表

]# grep ^root /etc/sudoers

root ALL=(ALL) ALL

root可以是%组名 ， 目标身份，省略时表示root

例：允许mike以root权限执行/sbin/下的所有命令，但是，禁止修改eth0网卡的参数

]# visudo

... ...

mike localhost,client50(主机名)=/sbin/\*,!/sbin/ifconfig eth0

示例：给用户bob、tom提权

[root@client50 ~]# vim /etc/sudoers

bob localhost,client50=/bin/systemctl \* mysqld,/bin/vim /etc/my.cnf

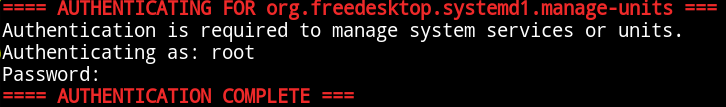
tom localhost,client50=/bin/systemctl \* httpd ,/bin/vim /etc/httpd/conf/httpd.conf

[root@client50 ~]# su - bob

[bob@client50 ~]$ sudo -l



[bob@client50 ~]$ systemctl restart mysqld //需要用户bob的密码



[bob@client50 ~]$ systemctl status mysqld

Active: active (running) since 一 2018-11-05 14:09:32 CST; 13s ago

[bob@client50 ~]$ systemctl stop mysqld //需要用户bob的密码

Active: inactive (dead)

[tom@client50 ~]$ sudo -l



[tom@client50 ~]$ systemctl restart httpd

Password: //输入tom用户的验证密码

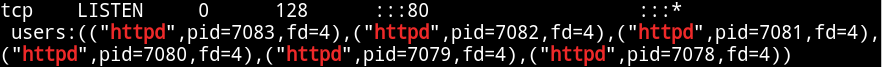
[tom@client50 ~]$ systemctl status httpd

Active: active (running) since 一 2018-11-05 14:15:56 CST; 7s ago

[tom@client50 ~]$ ss -natpul | grep :80

tcp LISTEN 0 128 :::80 :::\*

[root@client50 ~]# ss -natpul | grep httpd



[tom@client50 ~]$ sudo /bin/vim /etc/httpd/conf/httpd.conf

Listen 8909

[tom@client50 ~]$ ss -natpul | grep :8909

tcp LISTEN 0 128 :::8909 :::\*

[root@client50 ~]# ss -natpul | grep httpd



示例:给组提权

### - wheel组的用户无须验证可执行所有命令

[root@client50 ~]# groupadd dbadmin

[root@client50 ~]# groupadd webadmin

[root@client50 ~]# echo 123456 | passwd --stdin lucy

[root@client50 ~]# echo 123456 | passwd --stdin jeck

[root@client50 ~]# echo 123456 | passwd --stdin sarah

[root@client50 ~]# echo 123456 | passwd --stdin harry

[root@client50 ~]# useradd -G dbadmin lucy

[root@client50 ~]# useradd -G dbadmin jeck

[root@client50 ~]# useradd -G webadmin sarah

[root@client50 ~]# useradd -G webadmin harry

[root@client50 ~]# vim /etc/sudoers

#%wheel ALL=(ALL) ALL

%dbadmin localhost,client50=/bin/systemctl \* mysqld,/bin/vim /etc/my.cnf

%webadmin localhost,client50=/bin/systemctl \* httpd ,/bin/vim /etc/httpd/conf/httpd.conf

[root@room9pc01 ~]# ssh jeck@192.168.4.50

[jeck@client50 ~]$ sudo -l //输入jeck用户的验证密码

用户 jeck 可以在 client50 上运行以下命令：

(root) /bin/systemctl \* mysqld, /bin/vim /etc/my.cnf

注：用户jeck可以对数据库mysqld服务有管理权限

## 分析sudo提权的使用情况

### 修改全局配置，启用日志

- Defaults logfile=”/var/log/sudo”

[root@client50 ~]# vim /etc/sudoers

Defaults logfile="/var/log/sudo"

终端1：

[root@client50 ~]# tail -f /var/log/sudo

Nov 5 15:02:33 : bob : TTY=pts/1 ; PWD=/home/bob ; USER=root ; COMMAND=/bin/yum -y install elinks

终端2：

[root@room9pc01 ~]# ssh bob@192.168.4.50

[bob@client50 ~]$ sudo /bin/yum -y install elinks

### sudo别名设置

主要用途：

- 提高可重用性、易读性

- 简化配置、使记录更有条理

- 别名的名称必须全部大写

[root@client50 ~]# vim /etc/sudoers

Cmnd\_Alias SOFTWARE = /bin/rpm, /usr/bin/up2date, /usr/bin/yum

bob localhost,client50=/bin/systemctl \* mysqld,/bin/vim /etc/my.cnf ,SOFTWARE

[root@room9pc01 ~]# ssh bob@192.168.4.50

[bob@client50 ~]$ sudo -l

[bob@client50 ~]$ sudo /bin/yum -y install elinks

[bob@client50 ~]$ rpm -q elinks //提示软件包已安装

[bob@client50 ~]$ sudo /bin/rpm -e --nodeps elinks //卸载软件包

[bob@client50 ~]$ rpm -q elinks //提示未安装软件包

# SSH访问控制

## SSH基本防护

### 1. 概述：

### 存在的安全隐患

- 密码嗅探、键盘记录

- 暴力枚举帐号、猜解密码

### 常见的防护措施

- 用户限制、黑白名单

- 更改验证方式（密码——>密钥对）

- 防火墙 ... ...

### sshd基本安全配置

### 配置文件/etc/ssh/sshd\_config

### - Port 3389 //改用非标准端口

### - Protocol 2 //启用SSH V2版协议

### - ListenAddress 192.168.4.50

### - PermitRootLogin no //禁止root登录

- PermitEmptyPassword no //禁止密码为空的用户登录

- UseDNS no //不解析客户机地址

- LoginGraceTime 1m //登录限时

- MaxAuthTries 3 //每连接最多认证次数

## **sshd黑/白名单配置**

- DenyUsers USER1 USER2 ... //拒绝用户登录

- AllowUsers USER1@HOST USER2 ... //允许用户登录

- DenyGroups GROUP1 GROUP2 ... //拒绝组用户登录

- AllowGroups GROUP1 GROUP2 ... //允许组用户登录

注：以上条件只存其一，不能同时存在

### 应用示例：

- 仅允许一部分用户（从指定地点）登入

- 其他任何用户均禁止登入

]# vim /etc/ssh/sshd\_config

... ...

AllowUsers bob [root@192.168.4.254,192.168.0.110](mailto:root@192.168.4.254,192.168.0.110)

或者

AllowUsers tom jerry sarah

## SSH密钥对验证

### sshd验证方式控制

口令验证

- 检查登录用户的口令是否一致

密钥验证

- 检查客户端私钥与服务器上的公钥是否匹配

PasswordAuthentication no //密码认证

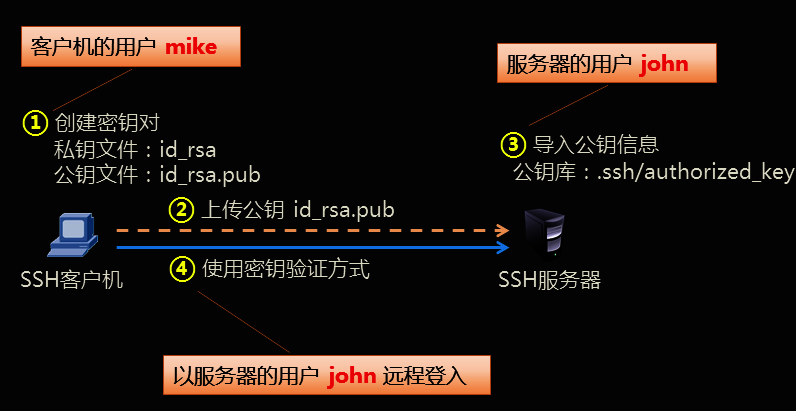
当把公钥文件传输给目标机器后，关闭本机上的密码认证登录，则只能使用密钥对登录，不能使用密码登录

PubkeyAuthentication yes

AuthorizedKeysFile .ssh/authorized\_keys

//公钥库：存放授权客户机的公钥文本

### 密钥对验证的实现思路



### 创建SSH密钥对

使用工具ssh-keygen

- 可以手动指定加密算法（-t rsa 或 -t dsa）

- 若不指定，默认采用RSA加密

### 部署SSH公钥

方法一，通过ssh-copy-id自动部署

- 优点：一步到位

- 局限：要求SSH口令认证可用

方法二，通过FTP等方式上传，手动添加

- 好处：灵活、使用范围广

- 局限性：操作繁琐、易出错

### 客户端登录操作

- 仅限密钥对创建人使用

- 须验证私钥口令（如果有的话，否则免密码登录）

示例：

要求：spec2作为服务器，spec1为客户端，spec3为自己的随身终端

客户端可以免密码ssh服务器spec2,随身终端亦可以连接服务器

步骤：

sepc1,sepc2创建用户zhangsan,tom

客户端生成密钥：

[zhangsan@spec1 ~]$ ssh-keygen

[zhangsan@spec1 ~]$ ls -Al /home/zhangsan/.ssh/

-rw-------. 1 zhangsan zhangsan 1675 11月 5 20:32 id\_rsa

-rw-r--r--. 1 zhangsan zhangsan 396 11月 5 20:32 id\_rsa.pub

将密钥传给服务器：

[zhangsan@spec1 ~]$ ssh-copy-id tom@192.168.4.2

[zhangsan@spec1 ~]$ ssh [tom@192.168.4.2](mailto:tom@192.168.4.2)

[tom@spec2 ~]$ whoami

tom

将私钥传递给spec3,通过spec1连接spec2

[zhangsan@spec1 ~]$ scp /home/zhangsan/.ssh/id\_rsa [root@192.168.4.58:/root/.ssh](mailto:root@192.168.4.58:/root/.ssh)

[root@spec2 ~]# vim /etc/ssh/sshd\_config

PasswordAuthentication no

[root@spec2 ~]# systemctl restart sshd

[root@spec3 ~]# ssh tom@192.168.4.2

[tom@spec2 ~]$ whoami

tom

# SELinux安全防护

## SELinux概述

### Security-Enhanced Linux

- 一套强化Linux安全的扩展模块

- 美国国家安全局主导开发

### SeLinux的运作机制

- 集成到Linux内核（2.6及以上）

- 操作系统提供可定制的策略、管理工具

## 红帽的SELinux策略集

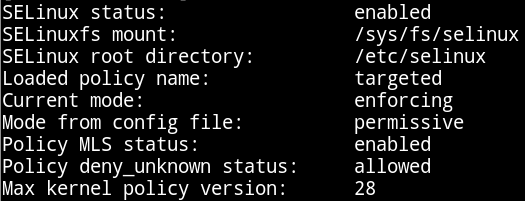
### SELINUXTYPE=targeted

- 推荐，仅保护最常见/关键的网络服务，其他不限制

- 主要软件包：selinux-policy、selinux-policy-targeted、libselinux-utils

libselinux-utils、coreutils、policycoreutils

[root@client50 ~]# sestatus



## SELinux模式控制

### 方法一，修改kernel启动参数

- 添加selinux=0以禁用

- 添加 enforcing=0设置SELinux为允许模式

### 方法二，修改文件/etc/selinux/config

- 设置SELINUX=disabled以禁用

- 设置SELINUX=permissive宽松/允许模式

- 设置SELINUX=enforcing强制模式

## SELinux开关控制

### 临时调整，setenforce命令（非Disabled状态下才可用）

设为1，对应强制模式

设为0，对应宽松模式

[root@client50 ~]# getenforce //查看当前状态

[root@client50 ~]# setenforce 1 //设置为强制模式

[root@client50 ~]# getenforce //确认结果为enforcing

## SELinux策略设置

### 查看安全上下文值

- 为文件/目录/设备标记访问控制属性

### 属性构成

- 用户：角色：访问类型：选项...

system\_u : object\_r : httpd\_sys\_content\_t : s0

## 修改安全上下文

### 使用chcon工具

-t ，指定访问类型

-R ，递归修改

### 一般操作规律

- 移动的文件，原有的上下文属性不变

- 复制的文件，自动继承目标为止的上下文

### 应用示例：

有一个权限为644的文件，但是FTP无权下载，分析故障原因，并解决

[root@client50 ~]# ll -Zd /var/www

drwxr-xr-x. root root system\_u:object\_r:httpd\_sys\_content\_t:s0 /var/www

[root@client50 ~]# ll -Zd /webroot/

drwxr-xr-x. root root unconfined\_u:object\_r:default\_t:s0 /webroot/

[root@client50 ~]# chcon -R --reference=/var/www /webroot

[root@client50 ~]# ll -Zd /webroot/

drwxr-xr-x. root root system\_u:object\_r:httpd\_sys\_content\_t:s0 /webroot/

[root@client50 ~]# echo bbbb > /root/b.html

[root@client50 ~]# mv /root/b.html /var/www/html/

[root@client50 ~]# ll -Z /var/www/html/

-rw-r--r--. root root unconfined\_u:object\_r:admin\_home\_t:s0 b.html

-rw-r--r--. root root unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 test.html

# chcon -t httpd\_sys\_content\_t /var/www/html/b.html

[root@client50 ~]# ll -Z /var/www/html/b.html

-rw-r--r--. root root unconfined\_u:object\_r:httpd\_sys\_content\_t:s0 /var/www/html/b.html

## 重要安全上下文

### 使用restorecon工具

- 恢复为所在位置的默认上下文属性

- -R ，递归修改

[root@client50 ~]# restorecon /var/www/html/c.html

[root@client50 ~]# restorecon -R /var/www/html/

### /.autorelabel文件

- 下次重起后全部重置

## 调整SELinux布尔值

### 使用getsebool查看

-a ，可列出所有布尔值

[root@client50 ~]# getsebool -a //列出所有的sebool值状态

[root@client50 ~]# getsebool -a | grep ftp //查看服务的sebool状态

### 使用setsebool设置

-P ，永久更改，重起后仍然有效

示例：打开samba的读写功能

[root@client50 ~]# setsebool -P samba\_export\_all\_rw on

安装vsftpd软件

[root@client50 ~]# yum -y install vsftpd

[root@client50 ~]# systemctl restart vsftpd

[root@client50 ~]# ss -natpul | grep :21

创建共享目录

[root@client50 ~]# mkdir /var/ftp/sharedir

[root@client50 ~]# chmod o+w /var/ftp/sharedir

[root@client50 ~]# ll -d /var/ftp/sharedir

[root@client50 ~]# cp /etc/passwd /var/ftp/sharedir/

[root@client50 ~]# chmod +r /var/ftp/sharedir/passwd

修改配置文件：

[root@client50 ~]# vim /etc/vsftpd/vsftpd.conf

29 anon\_upload\_enable=YES  **//去注释**

[root@client50 ~]# systemctl restart vsftpd

客户端连接：

安装软件包提供命令行连接工具

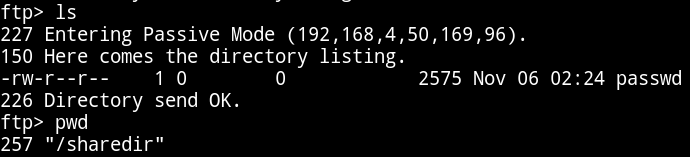
[root@spec2 ~]# yum -y install ftp

[root@spec2 ~]# ftp 192.168.4.50

Name (192.168.4.50:root): ftp

Password: //匿名用户密码设置为空

ftp> ls



ftp> cd sharedir //切换目录

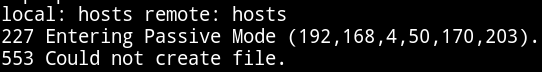
ftp> get passwd //下载文件

ftp> lcd /etc/ //切换到本地目录

Local directory now /etc

ftp> put hosts //上传文件

此时由于服务端selinux处于enforcing模式，所以上传文件失败

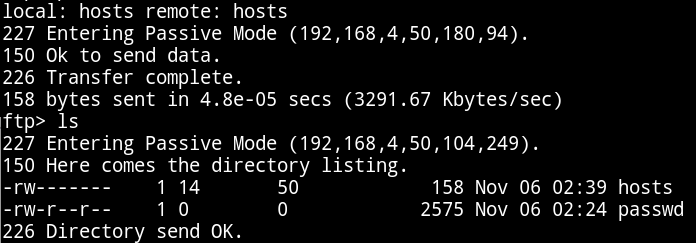


[root@client50 ~]# setenforce 0 //服务端更改selinux状态

[root@client50 ~]# getenforce

Permissive

ftp> put hosts //再次上传文件即可成功



修改sebool直接拥有上传权限

[root@client50 ~]# setsebool -P ftpd\_anon\_write on

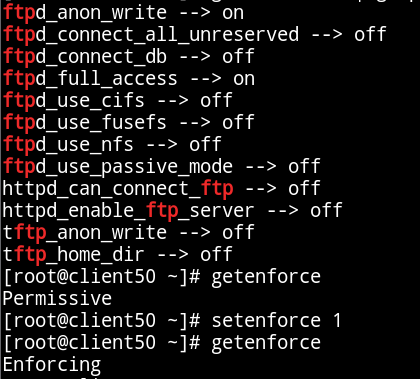
[root@client50 ~]# setsebool -P ftpd\_full\_access on

[root@client50 ~]# getsebool -a | grep ftp

[root@client50 ~]# getenforce //查看当前selinux状态

[root@client50 ~]# setenforce 1 //设置为强制模式

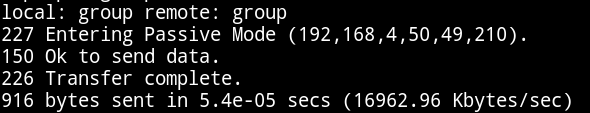
[root@client50 ~]# getenforce



ftp> cd sharedir //重新连接或者直接上传

ftp> lcd /etc/

ftp> put group //此时上传不受selinux的限制

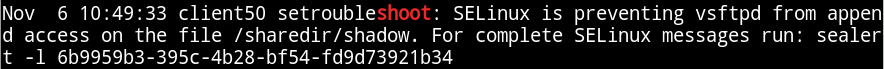


如何获悉要修改哪个selinux选项，提供上**传权限：**

[root@client50 ~]# rpm -qa | grep -i shoot

[root@client50 ~]# grep shoot /var/log/messages

监控软件的日志信息： /var/log/messages



]# sealert -l 6b9959b3-395c-4b28-bf54-fd9d73921b34

//执行run后面的命令查看详细信息

# chattr问题现象及解决

### 现象：为一个文件设置+a属性以后，无法用vim编辑的方式向文件末尾添加新内容

原因分析：使用vim修改文件会影响文件属性，这是+a不允许的

解决方法：改用>>追加的方式添加新内容

# ssh访问响应慢

问：客户机设置DNS以后，使用ssh连接远程主机时，在提示输密码之前要等待好久，清空/etc.resolv.conf设置则没这个问题

原因：sshd服务端会尝试查找客户端的主机名，GSS认证会尝试查找服务端的主机名

解决方法：

服务端启用UseDNS no,或者添加hosts映射

客户端启用GSSAPIAuthentication no ,或者添加hosts 映射